

Thesis Proposal:

Solution Based Problem Evolution and Problem Inception in Biologically Inspired Design

Michael E. Helms

Advisor: Dr. Ashok Goel

**Design and Intelligence Lab
School of Interactive Computing
Georgia Tech**

Last Updated: 08/18/2011

Thesis Proposal Committee

Dr. Ellen Do	School of Interactive Computing, Georgia Tech
Dr. Ashok Goel	School of Interactive Computing, Georgia Tech
Dr. Julie Linsey	Dept. of Mechanical Engineering, Texas A&M
Dr. Nancy Nersessian	School of Interactive Computing, Georgia Tech
Dr. Marc Weissburg	School of Biology, Georgia Tech

Abstract

Biologically inspired design is one of the most important movements in contemporary engineering design. The paradigm entails the use of biological systems as sources of ideas for the design of technological products. Thus, the paradigm, by definition, is based on cross-domain analogies. The movement is pulled by the growing need for sustainable development and pushed by the desire for creative design of innovative products. The design of wind turbine blades analogous to the tubercles on the fins of humpback whales is an illustrative example.

Despite its many successes, the practice of biologically inspired design remains ad hoc, with little systematization of either the design processes or of biological knowledge from a design perspective. Transformation of biologically inspired design into a principled design methodology requires the development of computational models, methods and tools. In the other direction, biologically inspired design provides a rich context for developing new computational theories of design, analogy and creativity.

To develop computational models and methods for biologically inspired design, we started with in situ cognitive studies of its practice. The results of these studies contained a surprise. While most design starts with a problem -- problem-driven design -- biologically inspired design often uses biological designs as analogues to identify and evolve design problems. We call this solution-based design.

The goal of my Ph.D. work is to develop computational methods and tools for solution-based problem evolution and inception in biologically inspired design. My thesis is that solution-based problem evolution and inception emerge out of the interaction of three core elements: 1) knowledge schemas that designers use to represent design problems and solutions, 2) goal-based information processes for evolving design problems, and 3) performance-based information processes for generating new problems. I will evaluate this thesis through detailed analyses of case studies of biologically inspired design, as well as through development of interactive tools for aiding problem inception and evolution. I will assess the interactive tools by deploying them in Georgia Tech classes on biologically inspired design. I expect my thesis to make two main contributions. On one hand, it will develop computational methods and tools for solution-based biologically inspired design. On the other, it will build computational models of solution-based creative analogies.

Contents

1. Introduction.....	6
2. Summary of Supporting Data	7
3. Research Agenda	8
3.1 Supporting Narrative	9
3.2 Content Account for Solution-based Design	11
3.3 Information Processing Account of Solution-Based Problem Evolution	12
3.4 Information Processing Account of Solution-Based Problem Inception	13
3.5 Supporting Tools and Infrastructure for Solution-based Design.....	14
4. Biologically Inspired Design	15
4.1 Biologically Inspired Design Class	15
4.2 Biologically Inspired Design Example: RaPower, 2009	16
5. Descriptive Data	18
5.1 Requirements documentation: Fall 2009.	19
5.2 Analysis of RaPower	21
5.2.1 Problem Schema.....	21
5.2.2 Data gathered.....	21
5.2.3 Summary Data	22
5.2.4 Solution Relationship Data	23
5.2.5 Final Analysis	24
5.3 Cases of solution-based problem inception: 2006, 2007	26
5.4 Solution-based problem inception: 60 Biologically Inspired Designs	27
6. Process Accounts of Biologically Inspired Design.....	28
6.1 Problem-driven and Solution-based design.	29
6.1.1 Problem-driven biologically inspired design	29
6.1.2 Solution-based Biologically Inspired Design Process.....	31
6.2 Compound analogy.....	32
7. Existing Accounts of Problem Evolution	34
7.1 Defining design as an ill-structured problem	34
7.2 Two schools of thought on problem structure.....	34
7.3 Top-down design	35
7.4 Solution-orientation	35

7.5 Problem Decomposition in Design	36
7.6 Time Spent on Problem Structuring	37
7.7 Problem-solution as co-evolution	37
8. Existing Accounts and Tool Support for Biologically Inspired Design.....	40
8.1 Process and Cognitive Support for Biologically Inspired Design	40
8.2 Support for Biologically Inspired Design.....	40
9. Hypotheses: Evaluation	43
10. Content Account for Solution-based Design	44
10.1 Hypothesis (1).....	44
10.2 Motivation	44
10.3 Assumptions	45
10.4 Method of Investigation & Evaluation	45
10.4.1 Method 1: Data Analysis.....	45
10.4.2 Method 1 Evaluation	45
10.4.3 Method 2: Tool Deployment	46
10.4.4 Method 2 Evaluation	46
10.5 Work-to-date	46
10.6 Remaining work.....	47
11. Information Processing Account of Solution-Based Problem Evolution	48
11.1 Hypothesis (2).....	48
11.2 Motivation	48
11.3 Assumptions	49
11.4 Method of Investigation & Evaluation	49
11.4.1 Method 1: Data Analysis.....	50
11.4.2 Method 1: Evaluation	50
11.4.3 Method 2: Tool-Based Experiment.....	50
11.4.4 Method 2: Evaluation	51
11.5 Work Completed	51
11.6 Remaining Work	52
12. Information Processing Account of Solution-Based Problem Inception	53
12.1 Hypothesis (3).....	53
12.2 Motivation	53

12.3 Assumptions	54
12.4 Method of Investigation and Evaluation	54
12.4.1 Method 1: Data Analysis.....	54
12.4.2 Method 1: Evaluation	54
12.4.3 Method 2: Tool-Based Experiment.....	54
12.4.4 Method 2: Evaluation	55
12.5 Work Completed	57
12.6 Remaining Work	57
13. Timeline for completion	58
14. References	60
15. Terminology.....	64
Appendix A: Problem-Driven Design Example.....	66
Sample problem-driven design project: i-Fabric	66
Appendix B: Solution-Driven Design Example	68
Sample solution-driven project: Abalone Armor	68
Appendix C: Compound Analogy Design Example	70
Example of Compound Analogy: Eye in the Sea.....	70
Appendix D: Instructor Requirements List	71
Appendix E: Problem Schema Specification	72
Artifact Specifications.....	72
Operational Environment.....	72
Function.....	73
Criteria	74
Problem Schema Relationships	74
Appendix F: Designer Problem Goals	76

1. Introduction

Biologically inspired design entails the use of a biological system as a source of design information for the design of a man-made artifact. The design artifact itself may be a complex mechanical, architectural, electronic or even informational system (to name a few). Alternatively, the artifact may be a material or technology useful for supporting the development of other complex systems. Perhaps the most common example of biologically inspired design is Velcro, inspired by the Burr plant.

In a senior level biologically inspired design class at the Georgia Institute of Technology, students use analogies to biological source systems to solve problems of their own invention; that is students generate the design problems they will solve. In a observations made over the last five years of this class, students were seen to spontaneously invent new problems inspired by a biological source, as well as to modify existing problems mid-design in response to a newly discovered biological source. Observations demonstrate that 100% of problems observed in class changed over time in response to existing biological sources, and almost 50% are so-called “solution-based designs” from inception (Helms, et al 2008). Instructors in the context of the biologically inspired classroom acknowledge that problem evolution is both difficult for students and is under-supported pedagogically and technologically.

Existing tools and theories of biologically inspired design focus on solving theoretical and practical problems inherent in interdisciplinary analogical design. Most propose a theory of source system representation, such as SBF (Bhatta & Goel 1996, Vattam et al 2010, Helms, Vattam & Goel 2010), SAPPhIRE (Chakrabarti et al 2005, Sarkar & Chakrabarti 2008), and function basis models (Nagel et al 2008, Shu et al 2007). Using these underlying models, systems deliver value by enhancing indexing, retrieval, and transfer of biological source systems to target engineering design problems. This is a central and valuable problem. However, systems currently do not provide a similar depth of support for representing design problems or for the process problem evolution in biologically inspired design. Existing theories and tools for biologically inspired design focus on solution development, and downplay the influence of biologically sources on problem development.

I will review the motivating data solution-based problem evolution and inception, discuss the current theory on design problem evolution, and review the current state of the art in biologically inspired design technology support. This will expose the underlying theoretical and technology gaps, motivating the thesis. I will then propose and motivate three hypotheses related to the thesis, and provide specific methods for investigation and validation. I will close with a proposed timeline.

2. Summary of Supporting Data

In the context of biologically inspired design class projects, students are challenged with creating conceptual designs for ill-structured problems that they generate themselves. Not only do student designers invent solutions, but they invent the problem to solve as well. While solution methodologies and scaffolding exists to assist the development of solutions, there is little scaffolding available to assist the development of problems. Problem development in this context is largely ad hoc and opportunistic, particularly for inexperienced design students.

In interviews I conducted in 2009 with the three primary instructors of the class, where the purpose of the interview was to understand how software could best be deployed to assist student designers, almost one-third of the requirements focused on the understanding, articulation or evaluation of the *design problem* the students were working on. For example instructors wanted to have students “define design sub-problems as they occur.”

In a study I conducted in 2010 evidence shows a high degree of change in problem descriptions over the course of the design trajectory. In the RaPower case study, over four iterations as many as 60% of concepts associated with a problem were dropped from iteration to iteration; only 10% of the original problem concepts carried through from initial to final design. The RaPower case study also shows approximately 80% conceptual turnover with respect to proposed solutions.

Despite the seemingly ad hoc and opportunistic approach students take, close examination reveals underlying patterns of conceptual use and transfer in the development of problem. In particular, even prior to the existence of a proposed *new solution*, evidence shows that *existing solutions* strongly influence the conceptual development of problems, for example in the RaPower study, as many as 60% of the functional specifications are derived from existing solutions. The observation that existing solutions influence problem conceptualization is empirically supported in this work, and remains an open problem for existing theories of design cognition.

An interesting characteristic of biologically inspired design is that a new design can result not only from an initially conceived of problem, but also by beginning with a biological solution. This so-called “solution-based design” pattern accounts for almost 50% of the 2006 and 2007 class projects. Additionally many of the most successful commercial designs -- Velcro and Lotusan Paint to name the most prominent -- also appear follow this design pattern. In fact, in an analysis of 60 biologically inspired designs (Vattam, Helms & Goel 2007), almost 80% of those that are either currently in the market or developed as marketable prototypes appeared to be solution driven. That the solution-driven process accounts for the vast majority of practical, real-world biologically inspired solutions indicates the value of achieving a deeper understanding of this process. Yet, while solution-based design inception appears quite common in biologically inspired design, no current theory of design explicitly accounts for this phenomenon.

3. Research Agenda

In this section I review the problem my research investigates, including the thesis and the set of hypotheses it drives. Prior to defining individual hypotheses, I will provide a *supporting narrative* for the problem so that readers may situate the thesis and individual hypothesis in the broader context. In subsequent sections I will elaborate on each hypothesis in greater detail.

Problem: The phenomena of solution-based problem evolution and problem inception are documented and valuable design processes.

Existing theories and tools use existing solutions for the development of new solutions, rather than problem development, and thus do not provide detailed explanatory accounts of the phenomena.

Thesis: Solution-based problem evolution and inception emerge out of the interaction of three core elements:

- 1) Four different schemas that designers use to represent design problems and solutions,***
- 2) Specific goal-based information processes and problem and solution schema interactions to evolve existing problems, and***
- 3) Specific performance-based information processes to generate new problems.***

3.1 Supporting Narrative

The background data and a number of descriptive accounts of biologically inspired design (solution-based analogy, compound analogy) lead to the insight that problem descriptions generated by student designers are strongly influenced by the solutions they observe in the course of their research and classroom experiences. I shall often use a short-hand notation for this observed process, and simply call it “the process of interest.” In the background data section of this proposal I will more deeply ground this process using case study data and the descriptive cognitive accounts already developed.

The goal of my dissertation is to provide an explanatory account to match the descriptive accounts of the process of interest. The explanatory account will provide one possible model for understanding why certain solutions (and not others) influence design problem descriptions, and why certain aspects of those solutions are transferred to the problem description (and not others). This will take the form of an information processing account that will explain both why and how the process of interest takes place. This account will then be used to provide scaffolding for student designers in the biologically inspired design process.

My thesis, the explanatory account, is that solution based problem evolution and inception emerge out of the interaction of three core elements:

- 1) Four different schemas that designers use to represent design problems and solutions,***
- 2) Specific goal-based information processes and problem and solution schema interactions to evolve existing problems, and***
- 3) Specific performance-based information processes to generate new problems.***

To develop the explanatory account, I will first provide a content account of the process of interest based on observations and data gathered in biologically inspired design, and borrowing from engineering design theory. The content will contain four schemas: the goal schema, the problem schema, the solution schema, and a problem-solution schema. The schemas are limited in scope to the process of interest in the context of biologically inspired design in the classroom. Using this content account, and a number of case-studies, I provide a procedural account of how, when starting from a problem, one can use a long-term memory of solutions to further describe a problem in the context of certain design problem understanding goals. This forms the information processing account of solution-based problem evolution (or SBP-Evo) of the process of interest. Next, using the same content account, I will provide a modified form of the information processing account to explain how, starting with a solution, one might conceptualize a problem inspired by it. This forms the information processing account of solution-based problem inception (or SBP-Inc).

As a means of validating both the content and information processing accounts, I will build three supporting tools. The first tool provides a framework and methodology for the formalization of problem descriptions provided by the content account to help students describe problems in a way that provides more complete problem descriptions for biologically inspired design. Our experience suggests that these more complete problem descriptions will yield better overall solutions. The second tool, when provided a problem description (as represented by the first tool), provides a means of retrieving relevant solution-problem pairs from a database of solution-problem pairs. The retrieved solution-problem pairs will influence the evolution of the problem description in predictable ways, depending on the initial state of the problem. The third tool, a modified version of the second, when provided with a biological solution, provides a means of framing the biological solution to identify and structure a relevant problem that can be solved by that solution. Each tool will be based on the content and information processing accounts described earlier. Evaluation of the tools will be described in subsequent sections.

The models and tools for biologically inspired design created as a result of this research will provide value to three core stakeholders: the teachers of biologically inspired design, the at-large community of biologically inspired designers, and the scientists interested in the process of design by analogy. For teachers, the theories generated by this work will provide insight into how to best structure problems and biological solutions to maximize productive problem inception, problem evolution and analogy making. This research will also provide technology for the systematic deployment of such structured representations. For the larger biologically inspired design community, the theories generated by this work will make explicit the most successful, yet mostly undocumented, process of solution-based design, enabling biologically inspired designers to attend to the design steps for successful solution-based design. Finally, for the scientific community, especially those involved in the process of design by analogy, the theories generated by this work provide a new explanatory account for the understudied portion of design by analogy that is concerned with the production and evolution of design problems through analogy.

The following is a summary of the hypotheses I will investigate for my dissertation. Subsequent sections will provide more detailed descriptions of assumptions, hypotheses, research methods, and evaluation.

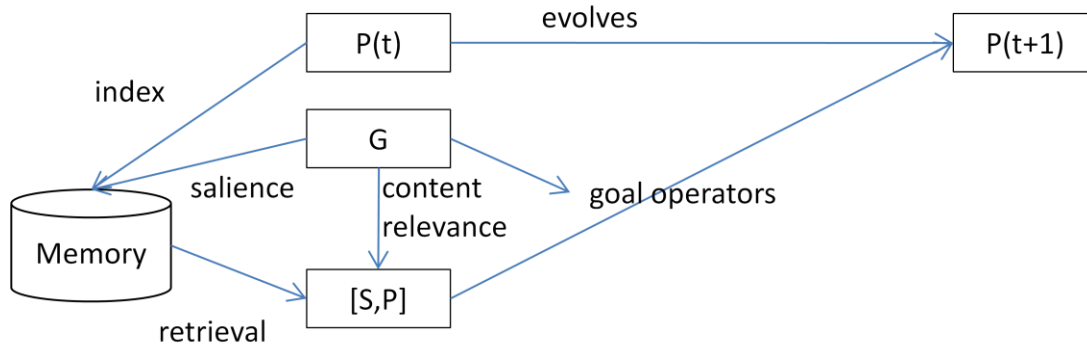
3.2 Content Account for Solution-based Design

Hypothesis (1): Designers in biologically inspired design use four different schemas in the process of design problem understanding. These schemas (a) generalize to problems and solutions in biologically inspired design, (b) enabling consistent mapping from observed conceptual elements to a schema element, and (c) provide a sufficient content account to explain the process of interest. The four schemas are:

- (1) Design designer problem goals (DPGs) account for goals relative to understanding the design problem itself including: conceptual addition, decomposition, partitioning, reorganization, prioritization and constraint relaxation. Each goal thus specified has a beginning state and end state, as well as a set of operators for moving from the beginning to the end state.
- (2) Design problem models in terms of a problem-schema (P-Schema) account for observed data, including operational environment, function, artifact specifications, and evaluation criteria.
- (3) Existing solution models in terms of a solution schema (S-Schema) account for observed data about structural specifications, function, behavior relative to an environment, and performance information.
- (4) Solution-problem [S, P] pairs account for the existing solution models in long-term memory that are associated with one or more P-Schema models in long-term memory (i.e. a solution is a solution to one or more problems).

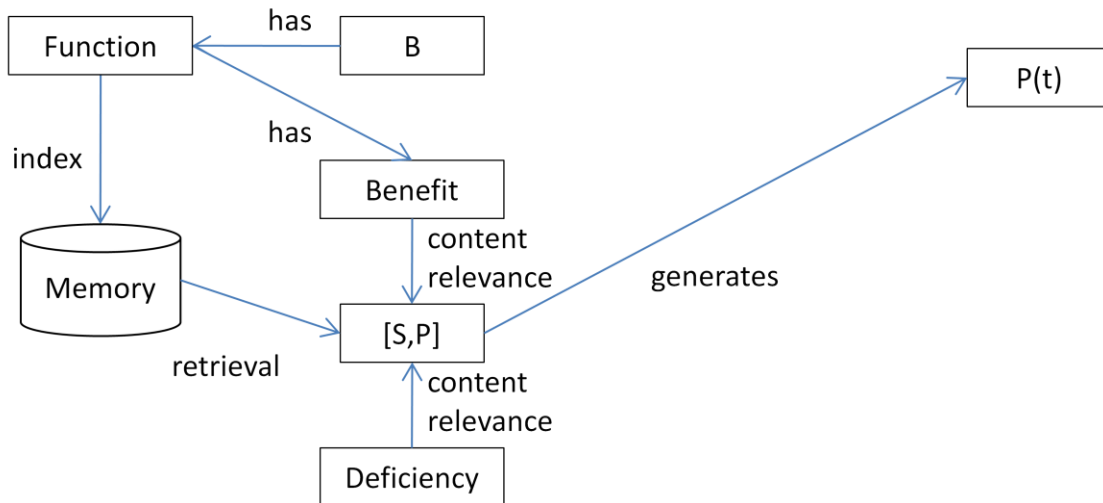
3.3 Information Processing Account of Solution-Based Problem Evolution

Hypothesis (2): Given a starting problem $P(t)$, and a designer problem goal G , one method of problem elaboration is by indexing, accessing and transferring problem concepts from a solution-problem $[S,P]$ pair in memory. The particular designer problem goal G , determines the indexing salience, transfer operators and content transferred. The process yields a new problem description $P(t+1)$.



3.4 Information Processing Account of Solution-Based Problem Inception

Hypothesis (3): Given a starting biological solution (B), a solution-problem [S, P] pair can be found in memory with either a structural, functional or environmental similarity (function shown in diagram) matching an index provided by the biological solution. If the performance of portion of the solution in the [S, P] pair is inferior in some way to the performance of an element of the biological solution, a new problem (P) can be generated from the problem specification provided in the [S, P] pair, and from the relative deficiency between matching elements in the biological solution (B) and the existing solution (S).



3.5 Supporting Tools and Infrastructure for Solution-based Design

The following tools will be developed in support of testing the above hypotheses. For all supporting tools, the context is student designers in a biologically inspired design context (either a class or project).

Supporting Tool (1) - I will provide a tool for the development of design problems in terms of the problem schema (P-Schema).

Goal (1-1): *Validation of Hypothesis 1*. Validate that the P-Schema is both sufficiently general, and sufficiently precise, for representing problems in the context of biologically inspired design class.

Supporting Tool (2) - I will create a shared technological environment for a searchable database of solution-problem pairs and a problem description framework and process (per Supporting Tool 1).

Goal (2-1): *Validation of Hypothesis 2: indexing and retrieval*. From some problem description $P(t)$, student designers can be proactively supplied with a solution-problem pair $[S,P]$ that is highly salient to $P(t)$, at time (t) .

Goal (2-2): *Validation of Hypothesis 2: problem evolution*. From some problem description $P(t)$, when students are supplied with some solution-problem pair $[S-P]$ that is highly salient, the problem $P(t)$ will change to problem $P(t+1)$ using information supplied from $[S-P]$, using the goal-based strategy dictated by a DPG.

Supporting Tool (3) - I will extend the shared technological environment in Supporting Tool (2) with a solution description tool and templates and processes for solution-driven design.

Goal (3-1): *Validation of Hypothesis 3: indexing and retrieval*. From some biological solution description (B) , student designers identify relevant solution-problem pairs using structural, functional or environmental similarity to (B) .

Goal (3-2) *Validation of Hypothesis 3: problem inception*. Given (B) and a matching $[S, P]$ students generate a new starting problem $P(0)$, based on the relative performance disparity between B and $[S, P]$.

4. Biologically Inspired Design

It is important to spend a small amount of time explaining the research context in more detail. Biologically inspired design is an important and growing movement in design (e.g., Bar-Cohen 2011, Bonser & Vincent 2007; Yen & Weissburg 2007, Benyus, 1997). The movement is driven in part by the need for environmentally sustainable development, and partly by the recognition that nature can be a powerful source of inspiration for technological innovations. Common examples of biologically inspired design include: fasteners (Velcro) inspired by the burr plant, dirt repellant paint inspired by the lotus plant, more efficient and quieter wind turbine blades inspired the fins of whales, etc.

The process of biologically inspired design entails the use of a biological system as a source of design information for the design of a man-made artifact, making it by definition analogical. Because such biological sources are radically different from the domain to which they are applied, the process and products from these analogies are more easily observed. Often multiple analogies are seen over the course of an entire design trajectory (Helms, Vattam, & Goel 2008, Helms, Vattam, & Goel 2009). Such prolific and easily identifiable use of analogy makes biologically inspired design an ideal domain for the study of the underlying processes and mechanisms of design by analogy.

4.1 Biologically Inspired Design Class

The rapid growth and interest in the field of biologically inspired design is driving the development of educational courses for supporting biologically inspired design in practice. Georgia Tech's Center for Biologically Inspired Design (<http://www.cbid.gatech.edu/>), offers a senior-level interdisciplinary course on biologically inspired design, and is planning undergraduate and graduate curriculums in the emerging interdisciplinary. According to Yen, et al (2011), "The connection between engineering and biology provided by BID as a problem solving activity provides an excellent atmosphere in which to encourage interdisciplinarity and develop sound pedagogical practices."

Since 2006 we have conducted multiple in situ cognitive studies of design teams in the biologically inspired design class. These studies have already led to the advancement of several new cognitive frameworks for biologically inspired design, and have significantly influenced how the class is structured over the last five years. For instance, problem-driven vs. solution-based design (Helms, et al 2008) are now taught as separate processes in class, and at least one of each type of project is required for each design team. Compound analogy (Helms, Vattam, and Goel 2008) is also an explicitly taught technique and the organizing framework of structure-behavior-function (Bhatta & Goel 1996) is embedded throughout classroom exercises, homework and design reports.

The course itself is structured into lectures, found object exercises, and a semester-long design project. Most lectures are focused on exposing student designers to specific case studies in BID, while found object exercises require designers to bring in biological samples and to analyze the solutions employed by these samples. The semester-long design projects start by grouping an interdisciplinary team of 4-6 students together based on similar interests. Instructors ensure that each team has at least one designer with a biology background and a few from different engineering disciplines. In some class iterations, teams were allowed to select any design problem, in others, the class was focused around a more general design domain within which teams had to find a problem to work on. Each team was to research their problem and design a solution using biologically inspired design, based on one or more biological systems. Each team has one or more faculty as mentors who give expert advice when needed. All teams present their problem and initial design concepts during the middle of the term, then submit final designs during the last two weeks of class along with a final design report. Yen, et al (2011) describe the pedagogy of ME/ISyE/MSE/PTFe/BIOL 4740 in much greater detail.

4.2 Biologically Inspired Design Example: RaPower, 2009

The following provides just one example of biologically inspired design from the 2009 class. I select this example, as many findings from data analysis in the following sections are based upon it. The team was formed by the instructors based on feedback from student preferences, and consisted of one biologist, one mechanical engineer, one industrial engineer, and one chemical engineer, and was provided with a

focus on the problem of energy in the context of sustainable housing. This focus led eventually to a biologically inspired color changing cover for solar thermal water heaters to prevent overheating. The description of the design trajectory provided here is based on the analysis of four homework assignments turned in sequentially over the course of the semester approximately 2 weeks apart: a description of the problem; a midterm presentation; a second problem description; and the final presentation.

The team began with the initial open-ended problem of sustainably generating power. After an initial meeting, the team produced a range of sustainable types of energy – wind, solar, water, geothermal – discussing solutions such as wind turbines, photovoltaic cells, towers of liquid sodium heated through reflected light, chemical batteries, and storage of energy for later use using compressed air. The design team also mentioned fat as a means of storing energy in biology. Cost was highlighted as a salient constraint on their design. The designers also ranged extensively in discussing different places in which the current technologies were used: from coastal areas, to farms and cities; they also discussed relevant weather conditions, such as the amount of wind or sun, and extreme conditions such as those found in Death Valley. Criteria were vague, of the character “more efficient” or “costs less”.

The midterm presentation limited discussion of existing technological solutions to photovoltaic cells and coal plants, however a wide range of biological sources were considered including the desert snail, diatoms, photosynthesis, enzyme reactions, and the lotus leaf. High-level descriptions of the relevant functions of each biological source were described, for example that the function of the desert snail is heat dissipation, performed by the structure of its shell. The designers proceeded with proposing simple one-to-one corresponding solution-modifications to the photovoltaic cell, derived from each of these biological solutions. Thus, in the case of the self-cleaning lotus leaf, students proposed a self-cleaning photovoltaic cell. Solution proposals were little deeper than a function-solution pairing of the type just mentioned, none of which were developed further. From initial description to midterm, we notice the addition of new functions, cleaning-self and dissipating heat which were directly associated with biological solutions having the same functions. We note that designers drop other heat related functions, such as storing and directing heat. We also note that while the mirror/heat-tower solution is dropped, the environment in which it operates, the desert, remains in place, and is also the same environment in which the desert snail lives. Furthermore, the criteria “passively” is now associated with the heat dissipation and self-cleaning functions that were attached to biological solutions. Manufacturing also is a rising concern, as the ability to reproduce materials and effects is highlighted.

The second problem description assignment continues its focus on solar panels and photovoltaic cells. All of the biological sources mentioned previously, are maintained, except diatoms which appear to have been dropped. Heat dissipation is discussed, but the design team now focuses on a flexible, moldable and self-cleaning surface derived from the lotus leaf, and on a newfound perceived deficiency in current solar panels – rigidity. Furthermore, the operational environment has shifted from a desert focus, to a more dynamic environment with greater temperature range. As well, the team is focused on the need to connect their solution to a home

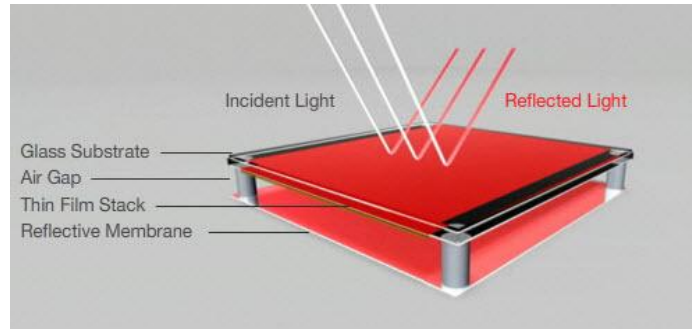


Figure 1. RaPower final solution.

(part of the initial design requirement). Again, students raise manufacturing of nanoscale materials as a concern, as well as the need for materials to be sustainable. The criteria focus has shifted from passive response in the midterm presentation to increased efficiency.

Figure 1 provides a graphic of the final design. Students arrive at a new solution, which is concerned with regulation and cooling, rather than self-cleaning and flexibility. The design team appears to have radically evolved the problem, now no longer working with photovoltaic cells, but looking at solar thermal collectors for water heating, which run the risk of overheating and damaging their internal structure. The solution is a dynamic feedback regulation mechanism from enzymes discussed in the midterm, combined with a solution inspired from a new biological organism, the tortoise beetle, which has a shell which it uses for camouflage by changing color. The designers intend to use a mechanism similar to the tortoise beetle to alter the color of the thermal collectors to change the amount of heat captured, depending on the internal heat of the unit. The final design, the SolShield, is the first solution generated as more than a simple function-solution concept.

While this final problem appears to be a new problem, we can see, in fact, the derivative nature of the process. Reacting to heat has been embedded in the teams thinking all along, from the mirror/heat tower, to the desert snail, to the operational environment of the desert, to the concept of dynamically responding to the environment. These concepts were influenced by a number of previous solutions that were investigated so that when a new problem concept arose – overheating -- the team was able to pivot to the new problem focus and quickly come up with a dramatic, creative solution.

5. Descriptive Data

In this section I describe in more detail the data gathered since 2006 that grounds the theory of solution-based problem evolution, and solution-based problem inception as relevant and important theories in biologically inspired design. This descriptive data demonstrates the two underlying phenomena that my explanatory model must fit. I will present the data thematically, rather than the order in which it was acquired.

The first of the two sets of descriptive data are focused on understanding solution-based problem evolution in biologically inspired design. The first data set anecdotally demonstrates that problem evolution in biologically inspired design is a difficult process for student designers to master. The second data set demonstrates that the process of problem evolution is strongly influenced by existing solutions.

The second of the two sets of descriptive data provide a strong argument for solution-based problem inception. The first data set demonstrates that solution-based problem inception is a naturally occurring phenomenon in the course of student design projects. The second data set demonstrates that solution-based problem inception occurs frequently outside of the classroom setting, and indeed appears to be the dominant form of successful biologically inspired design in real-world design scenarios.

5.1 Requirements documentation: Fall 2009.

In late Fall 2009, I interviewed three instructors and four students from the biologically inspired design class at Georgia Tech. The purpose of the interviews was to understand how software and technology might be deployed by the Design Intelligence Laboratory to assist students in the process of biologically inspired design. The interviews were conducted in part to gather requirements for the Design Intelligence Lab for future research and software deployments in the BID Classroom. At the time, the Design Intelligence Lab was focused on providing search support technologies for biologically inspired design, using a system called DANE, which was introduced into the classroom in 2009. The complete set of instructor requirements gathered during these interviews is provided in Appendix D: Instructor Requirements.

Each requirement was labeled with one or more of six categories which relate to the support type implied by the category label: analogy, communication, design support, problem, search, and technical. Only one requirement had more than one label. As shown in Table 1, instructors generated twice as many problem-support requirements than any other type (students were more focused on design support for rendering).

The Table 2 provides the list of instructor based requirements, which demonstrates instructor desire to support students in the articulation, understanding, evaluation and evolution of their design problems. My interpretation of this data, supported by my experience in the class, is that in the context of ill-structured biologically inspired design problems, student designers spend a great deal of time struggling with the definition and evolution of their design problem over the design trajectory. While some few teams are lucky enough to select a well-defined problem that is well understood by the team from the beginning, the vast majority of teams *spend as much time determining and changing the design problem they are trying to solve as they do generating solutions for it*. Instructors see this struggle, and desire a system to structure and more quickly evolve the design problem. The next set of data provides a much clearer description of the kinds of changes of the problem description that students deal with over a complete design trajectory.

Table 1. Instructor and student requirements by requirement type.

Category	Instructor	Student
Analogy	3	0
Communication	6	4
Design Support	7	11
Problem	14	5
Search	2	5
Technical	2	1

Table 2. Instructor Generated Problem Support Requirements

Help students better define problem

Ask students questions about their problem.

Force students to define problem in terms of final results

Define design sub-problems as they occur

Create traction for vaguely defined, intractable problems.

Assist students in problem partitioning

Assist students in evaluating problem partitions

Define measures of importance for problem partitions

Define problem (not solution) inputs and outputs.

Provide a means for understanding problems at different levels of abstraction

Provide a means to describe for natural systems the problems in nature they are solving

Allow for problems to be changed to better match solutions

Catalogue problems to solve

5.2 Analysis of RaPower

The RaPower design trajectory was presented as an example of biologically inspired design. My objective in this section is to provide an analysis for the evolving problem model in the RaPower case study. Problem concepts are mapped to a high-level problem schema and associated with other concepts to which they are explicitly linked in the data. The schema used is discussed in detail in Appendix E: Problem Schema Specification.

I was interested in understanding the connections between solutions and the problem schema. Thus I also required a solution description and a means for relating the two. For this I leveraged Structure-Behavior-Function modeling, a solution modeling schema already created and vetted in biologically inspired design.

5.2.1 Problem Schema

(O) Operational Environment	(F) Function
(A) Artifact Specification	(C) Criteria

For this analysis, I used the four high-level problem modeling schema shown in figure 2. I considered (1) specifications for the artifact itself, (2) the environment within which the artifact must operate, and (3) the intended functions of the artifact. The fourth dimension is the set of relative (compared to existing solutions) and absolute (not grounded to existing solutions) criteria imposed on (1), (2) and (3). Appendix E provides additional details on the definition of each category.

FIGURE 2. High-level Problem Schema

5.2.2 Data gathered

Four data points were used in this case study to track the progression of the problem over time. The first data point was a problem description assignment, the second a midterm presentation, the third another problem description assignment, and the fourth was the final design presentation.

Only the text content, including bullet points, formulae, tables and text annotations, from each of the four design documents were considered in this study. Text was divided into phrases, each of which encapsulated a single problem or solution schema concept. Some concepts, such as referencing an existing solution, are short and straightforward, such as “the desert snail”. Other concepts such as “so that it is cooler within the shell than the outside air and ground” are more verbose, but encapsulate essentially a single concept, in this case a criteria for the degree to which the function cooler must perform.

Table 3: Encoding Statistics				
	PD1	Midterm	PD2	Final
Concepts	124	120	116	106
Comments	13	2	13	7
Relevant Concepts	111	118	103	99
Ambiguous	10	1	8	5

Relationships were inferred directly from text. If a solution was mentioned with respect to an aspect of a problem concept, that solution was tagged to the problem concept. In this way, we could determine the linkage between a particular problem concept and a solution. For instance, in the phrase “we typically think of voltaic cells

creating current”, voltaic cells are an existing solution, and creating current is a function of that solution.

Table 3, shows the number of concepts encoded for each of the four data points, as well as the number of designer comments (excluded from this analysis), relevant concepts (total minus comments), and ambiguous encodings, which accounted for roughly 5% of the total number of concepts that were encoded.

5.2.3 Summary Data

Table 4 shows summary statistics for each of the four data points, across the four problem schema concepts. We include the summary number of existing and design solutions cited at each point. At this first level of analysis, some things already stand out. First, the number of functions considered at each stage remains between 20 and 25 until the final design, where it drops to 9. Second, the number of criteria, 25% of which were with respect to efficiency, was consistently high (between 10 and 17) for the first three data points, and decreases to only a few (3) at the final presentation. This seems to parallel the reduction in functions considered in the final presentation. Third, the number of artifact specifications is very low, never more than 6. Of the 14 total, four were cost-related and three were sustainable materials related. Fourth, the number of operational environment concepts initially considered was very large (23), but this was rapidly reduced to 4 by the second data point, and then gradually expanded throughout the remainder of the design. It appears that the designers ranged across many disparate environments initially, but then settled into an environment, which gradually took on finer levels of detail. Finally, the number of existing solutions referenced was consistent, between 5 and 9, trending down to 5 at the final presentation. The number of new solutions discussed moves from one to seven back to a single final new solution, the SolShield, in the end.

	PD1	Midterm	PD2	Final
Operational Environments	23	4	6	12
Functions	25	20	20	9
Artifact Specifications	1	6	3	4
Criteria	10	10	17	3
Existing Solutions Cited	9	8	6	5
Design Solutions Cited	1	7	3	1

With respect to functions, Table 5 considers the follow-through of each function from description to description. We see that from the initial generation of 25 functions, three of those functions carry forward into the Midterm, seven are considered in PD2, and two (“generate energy” and “capture energy”) follow through to the final. Likewise 17 new functions appear in the Midterm description, one of which (“adjust flow”) appears in the final design. Fewer new functions appear in the third data point, just 10; of which 1 (“keep cool”) makes it into the final model. In the final model itself, there are more new functions than old. Five new functions appear in the model, while four have been carried through from previous descriptions. This alone tells a very interesting story. In this ill-defined design problem, we see a great deal of exploration. Fifty-seven unique functions are considered, only nine of which eventually make it into the final solution. Over 80% of the functions considered are discarded along the way.

	PD1	Midterm	PD2	Final
PD1	25	3	7	2
Midterm	--	17	3	1
PD2	--	--	10	1
Final	--	--	--	5
Total	25	20	20	9

A second aspect to the function-specific data is the carryover of sub-concept level notions, in particular the carryover of the verb and function-object components of the functions. We note that while there were fifty-seven unique functions, these functions were comprised of combinations of only 24 verbs and 15 function-objects. Table 6 shows the carryover of just the verbs. One can see from this table that while it appears from a function perspective that 80% of the functions are discarded, we see heavy re-use of the verb sub-concepts, which is recombined with function-objects in later stages to formulate new functions. In fact, no new verb concepts are added in the final design, and only one new function-object is added. The “generation” of the five new functions in the final design comes from the recombination of existing verb/function-object concepts.

Table 6: Verb Carry-over				
	PD1	Midterm	PD2	Final
PD1	10	2	5	4
Midterm	--	11	4	2
PD2	--	--	3	0
Final	--	--	--	0
Total	10	13	12	6

5.2.4 Solution Relationship Data

I consider one more level of detail in the data; the relationship of solutions to the concepts in the problem model. Recall that for any concept in the problem model, that concept may be associated with an existing solution, a biological solution or with a new solution concept. The following four tables show for each data point the numbers of operational environment, function, artifact specifications, and criteria concepts respectively and whether they are associated with (1) an existing solution, (2) a biological solution, or (3) no solution association.

Table 7: Operational Environment by Solution reference				
	PD1	Midterm	PD2	Final
Man made	9	0	1	9
Biological	0	5	0	0
No reference	15	0	5	10
Total	24	5	6	19

Table 7 shows that operational environment concepts were initially generated from existing solutions (9), but even more concepts (15) were not associated with any solution. We see the opposite occur in the midterm, where the only operational environment concepts were associated with biological solutions, such as “desert areas” and “the presence of the sun” where we find the desert snail. Interestingly, after this short period of mentioning biological source operational environments, we don’t see a single reference to them for the remainder of the design process. Instead attention is divided between existing solutions and new environments.

Table 8: Function by Solution reference				
	PD1	Midterm	PD2	Final
Man made	18	4	10	4
Biological	0	12	6	6
No reference	11	6	12	3
Total	29	22	28	13

The trends for function in table 8 somewhat mirror those for operational environment across the first two data points, but whereas operating environments inspired by biological sources disappear altogether we find that certain biological functions stay with the team, eventually comprising almost half of the functions mentioned in the final presentation, and 3 of 13 functions in the final design were taken directly from biological sources. (Note: since a function may be associated with more than one solution, the total number of function-solution pairs cited in this table (13) is higher than the actual number of unique functions (9) found in the final design).

The trend for artifact specification in table 9 with respect to existing and biological solutions is clear. Specifications are not associated with, at least not explicitly in this case study, other solutions. Many of the specifications were with regard to cost and sustainable materials, which were likely inferred from the design context of “sustainable housing” (note, there were no “customers” per se with whom designers could interact).

Table 9: Artifact Specification by Solution reference				
	PD1	Midterm	PD2	Final
Man made	0	0	0	0
Biological	0	0	0	0
No reference	0	6	3	5
Total	1	6	3	5

Finally, we observe the trends in criteria in table 10. We speculated initially that criteria would be driven by performance characteristics of existing solutions. Designer’s initial criteria proved to be very vague, for example “to design a more efficient solution”. In this case, “more efficient” is compared to the notion of an existing solution, but, since the problem appears so ill-defined initially, there is no solution against which to benchmark performance. Designers leave their initial criteria ungrounded to specific solutions. This trend reverses itself when designers have narrowed in their scope, and by the third data point 11 of 17 references are now grounded with respect to another solution. However, in the final design, such criteria practically disappear altogether. This may be a result of the rapid pruning of their final solution to a few functions in the final step of the process.

Table 10: Criteria by Solution reference				
	PD1	Midterm	PD2	Final
Man made	0	2	11	1
Biological	0	3	2	1
No reference	10	5	4	1
Total	10	10	17	3

5.2.5 Final Analysis

This case study does not contain a well structured design problem or even a design brief from which to begin. There is no organized decomposition of sub-problems, and no generation of solutions to be evaluated to inform the next problem iteration. What we see is a team with an open-ended design problem, struggling as much to define the problem as to generate a solution. The goal of my research is to determine in this real world environment for underspecified problems, why certain solutions are selected to inform the problem and why some information from those solutions is transferred while other information is not.

Our data suggests that the design team explores different aspects of the problem model; committing to few concepts rigidly, holding open possibilities until the right confluence of problem model and existing solution models emerge to form a cohesive pair. As opposed to generating new solutions to problems as they form, designers employ other strategies to generate problem concepts and enrich their problem model.

From this case study it is clear that using existing solutions is one method of understanding, or at least adding concepts to, an existing design problem. Designers appear to tentatively adopt problem aspects from existing solutions, in particular functional aspects, temporarily appending them to an overall problem model. This portrays a clear picture of what I am calling solution-based problem evolution (SBP-Evo).

5.3 Cases of solution-based problem inception: 2006, 2007

In 2006 the following data were observed for the nine design projects conducted in class. Table 11 details the project name, the number of biological sources used in the design, whether or not solution fixation occurred, and whether the project followed a problem-driven or solution-based design approach. For the projects in 2006, we see that 4 out of 9 projects were solution-based – that is, a biological solution of interest was identified first, and then a problem subsequently found which that solution could solve. Likewise in 2007, we see that 5 out of

Table 11. Problem and solution-based problems, 2006 & 2007

Class Projects 2006			
Project	Sources Used	Initial Fixation	Solution/ Problem
Bomb Detection	2	-	Problem
Traffic Routing	1	Ants	Problem
Body Armor	1	Abalone	Solution
Air Filtration	2	-	Problem
Visual Display	2	Morpho	Solution
U/W Stealth Vehicle	2	Copepod	Solution
Cell Phone Case	1	Abalone	Solution
Class Projects 2007			
Project	Sources Used	Initial Fixation	Solution/ Problem
Squid stealth vehicle	1	-	Problem
U/W cavitation gun	1	Pistol shrimp	Solution
Xylem Tunnels	1	-	Problem
Spider helmet	3	-	Problem
Cartilage cushion	1	Cartilage	Solution
Avoidance detection	2	Bat sonar	Solution
Anti-fouling catheter	1	Shark skin	Solution
Color changing car	1	Squid	Solution
Oil Spill Clean up	2	-	Problem
Heat sink fin	2	-	Problem

10 projects used the solution-based approach. I provide a complete

description of the process of solution-based design in the Section 7.1 of this proposal. A detailed design trajectory for the body armor project is provided in Appendix B: Solution-based Design Example.

This data shows that in the two initial years of our observations in the biologically inspired classroom, almost 50% of the class projects were solution-based.

Note, although our data spans 2006 through 2010 the processes of solution-based and problem-driven design began to be taught explicitly in the class. Solution-based and problem-driven designs became prescribed methods for certain projects, imparting too heavy a bias for their occurrence to be viewed as “naturally occurring”.

5.4 Solution-based problem inception: 60 Biologically Inspired Designs

In 2006 we recorded 41 citations used in the biologically inspired design class that included a total of 60 biologically inspired designs. These citations included biologically inspired design cases at different points in the design process, from understanding the theory of the biological processes involved, to theories that have been applied to successfully marketed products. In each case the citations explained the current state of the one or more biologically inspired designs. After reading the citation, I coded each biologically inspired design as either: applied (a product was actively marketed), in development (prototypes were being created), theory (basic science was still being conducted), or unknown (if the status was not clear). Furthermore, I found that in each case from the literature I could infer that the inception of the design concept was either solution-based (which assumes primary research was conducted on the biological source prior to a product design) or problem-driven (a problem was identified, and later a biological solution was found that addressed the problem). Table 12 summarizes the finding of this research in terms of the current status of the research and whether the research was problem-driven or solution-based.

Table 12. Problem-based vs. Solution-based real-world examples of biologically inspired design.		
Status	Problem-based	Solution-based
Unknown	0	23
Theory	1	8
Development	2	11
Applied	4	11

The number of solution-based research projects outside of the classroom is far greater than the number of problem-based research projects. Only 6 of the 28 projects in development or being applied in the real-world were problem-driven examples. This data strongly implies that solution-based design is not only real, but the dominant process used in biologically inspired design outside of the classroom. Even considering the possibility of selection bias, one cannot dismiss the sheer volume of 53 solution-based design concepts.

6. Process Accounts of Biologically Inspired Design

In this section I summarize two existing process accounts of biologically inspired which I co-authored in 2008. These process account theories are evidence-based, supported by two years of data accumulated through classroom observations. The process accounts serve two purposes. First they provide further evidence that supporting the assertion that existing solutions strongly influence evolving problem descriptions. Second they provide a benchmark against which to measure the accuracy of a theory used to explain biologically inspired design. Any explanatory theory must answer, within the scope of the theory, how and why these process accounts originate.

6.1 Problem-driven and Solution-based design.

My co-authors and I (Helms, et al 2008) observed the existence of two high-level processes for biologically inspired design based on two different starting points – *problem-driven* and *solution-based*. Here we use the term solution-based to describe a process that begins without a particular problem in mind, and where starting point for ideation is a biological source system. The term solution-based or solution-driven design has been used alternatively to describe design processes that propose solutions prior to a deep analytical phase (Krugar & Cross, 2006), and in biologically inspired design to describe reverse-engineering and application of a biological solution (Wilson, 2008) to a given problem.

6.1.1 Problem-driven biologically inspired design

As depicted in Figure 3(a), in a problem-driven approach, designers formulate a problem that serves as the starting point for subsequent problem solving. The pattern of problem-driven biologically inspired design follows a progression of steps, which is non-linear and dynamic in the sense that output from later stages frequently influences previous stages, providing iterative feedback and refinement loops.

- Step1: Problem Formulation
- Step 2: Problem Reframing
- Step 3: Biological Solution Search
- Step 4: Define the Biological Solution
- Step 5: Principle Extraction
- Step 6: Principle Application

Step 1: Problem Formulation

Designers were asked to find or invent a problem they care to solve and then were instructed to define their problem as a function. For example a group that began with the problem of preventing shark attacks on surfers defined their desired function as camouflaging a surfboard. Problem elaboration typically occurred throughout the design process, creating more refined functional requirements and constraints.

Step 2: Reframing the Problem

Designers defined problems in human terms such as protecting police or avoiding shark attacks. To find solution

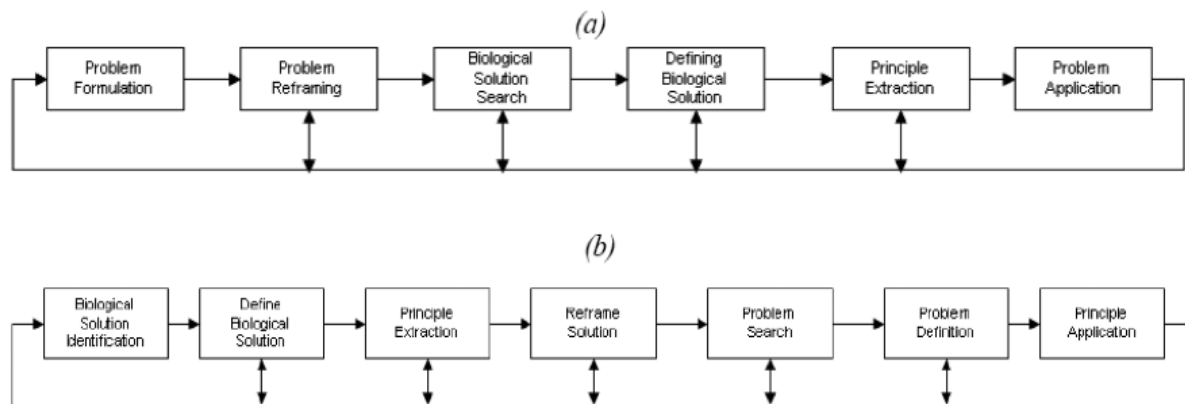


Figure 3. Observed biologically inspired design processes. (a) Problem-driven process. (b) Solution-based process.

analogues in biology, designers redefined their problems in more biological terms, often in the form of a question such as “How do biological solutions accomplish xyz function?” As an example, instead of “stopping a bullet,” the reframed version was “What characteristics do organisms have that enable them to prevent, withstand and heal damage?”

Step 3: Biological Solution Search

Instructors provided the following four general strategies for finding biological solutions.

Table 13 Solution Search Heuristics

Search Technique	Technique Description
Change Constraints	If the problem is narrowly defined, such as “keeping cool”, change the constraints to increase the search space, for instance to “thermoregulation”.
Champion Adapters	Find an organism or a system that survives in the most extreme case of the problem being explored. For instance, for “keeping cool”, look for animals that survive in desert or equatorial climates.
Variation within a Solution Family	Find organism “families” that have faced and solved the same problem in slightly different ways. For instance, the many variations on bat ears suggest deeper solution principles for echo-location.
Multi-Functionality	Find organisms or systems with single solutions that solve multiple problems simultaneously.

Step 4: Define the Biological Solution

Designers first identified structures and surface mechanisms from the biological system that were related to the reframed function, for example, the shell of the abalone for resisting impact. The initial understanding that abalone shell is hard, lightweight, resists impacts, and is regenerative, deepened over time into an understanding of the complex interactions of composite materials that are responsible for this behavior. We note that 66% of the all of design teams, and 100% of the design teams using the alternative solution-driven approach, focused on structure and surface characteristics in this way.

Step 5: Principle Extraction

After a solution was well understood, important principles were extracted into a solution-neutral form, which required a description that removed as many specific structural and environmental constraints as possible. For example describing the principles of the abalone shell in terms of “interactions between flexible proteins and hexagonal calcium carbonate deposits” may constrain design thinking to proteins, calcium carbonate, and hexagons. On the other hand “tightly coupled composite material formation with alternating flexible and rigid structures for resisting impact,” allows for the possibility of using arrangements of many different kinds of flexible and rigid material.

Step 6: Principle Application

After the principle was extracted from the biological solution, designers translated the principle into the new domain. This translation involved an interpretation from one domain space (e.g. biology) into another (e.g. mechanical engineering), by introducing new constraints and affordances. In the case of the bullet proof vest, new weight, flexibility, impact resistance and manufacturing process criteria were added, along with new affordances, for example in materials. This process often created new sub-problems, which designers frequently solved with new biologically inspired solutions. Designs that used multiple biological analogies we classify as *compound analogical designs* (Vattam et al, 2008).

6.1.2 Solution-based Biologically Inspired Design Process

Whereas the normative biologically inspired design process taught in the class was problem-driven, we observed that in practice the design process often began with a biological solution. Some classroom exercises, and many case-studies of biological design, began with a biological solution, extracted a deep principle, and then found problems to which the principle could be applied. In general, the solution-driven biologically inspired design process follows the steps listed below, as shown in Figure 3(b).

- Step 1: Biological Solution Identification
Designers start with a particular biological solution in mind.
- Step 2: Define the Biological Solution
- Step 3: Principle Extraction
- Step 4: Reframe the Solution
Reframing forces designers to think in terms of how humans view the usefulness of the biological function being achieved.
- Step 5: Problem Search
Whereas search in the biological domain includes search through some finite space of biological solutions, problem search may include defining entirely new problems.
- Step 6: Problem Definition
- Step 7: Principle Application

The process of solution-based design provides many clues about how problems and solutions might be organized in memory, and how they must interact with each other. As a result of the solution-based design process we know that a solution must have some “hooks” into problems; not just the problems the solution solves, but also the ability to access and modify other problems. Since this process is so heavily influenced by solutions, and since it represents so many of the observed cases of biologically inspired design, it seems natural to attempt to extend solution-based problem evolution to account for this process as well.

6.2 Compound analogy

Solving complex problems by decomposition where designers break complex problems into less complex ones is not new. But when we make the decompositions explicit in the context of analogical design, it becomes apparent that the processes of decomposition and analogy influence each other. We describe their interplay as *compound analogical design* (Helms, Vattam & Goel, 2008).

In the simplest case of compound analogical design, shown in Figure 4, when a target design problem is presented, the designer iteratively decomposes the problem into sub-problems to create a problem abstraction hierarchy. Our examples assume the problem is decomposed along functional lines, although we have observed other lines of decomposition (temporal, structural, etc.), often intermingled. Assuming that the problem is decomposed along functional lines, each node in this hierarchy is a function to be achieved. Each function (node) can be used as a cue to retrieve known solutions that achieve that function. Solutions are transferred to the current problem, and aggregated to generate the overall solution. This process explains complications that often arise during reintegration, as the solutions from disconnected analogies may not integrate cleanly at their boundaries, or may have overall constraint mismatches.

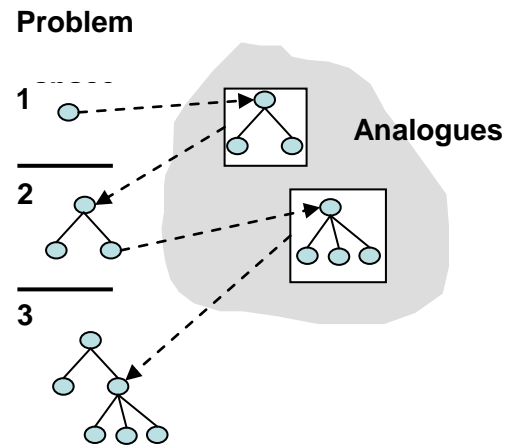


Figure 4. A simple case of compound solution generation.

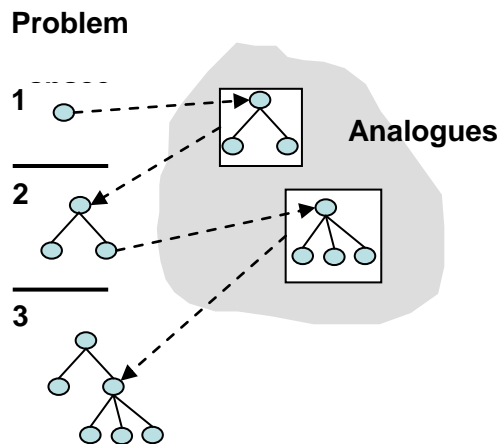


Figure 5. Iterative analogical generation of a problem space.

In many cases, it may not be obvious to the designer how to decompose a problem into manageable subparts. In this case, the designer might then search for an analogous solution based on the high-level problem itself. This retrieved analogical source not only provides a potential solution, *it may also allow the user to infer the problem decomposition in the source design*. This decomposition in the source design (along with solutions to the sub-problems) can be “brought into” the current problem space as shown in Fig. 5.

Each new node from the source solution decomposition integrated into the problem space can act as an additional cue for retrieving another set of solution analogues. This process can continue iteratively leading to the incremental development of the problem space. At every stage of this iterative process, the designer can evaluate the partial solutions available and can decide to take further actions. The apparently simple iterative feedback between these two processes provides a flexible problem solving framework that accounts for

the incremental evolution of complex, compound analogical design solutions. An example of compound analogical design is presented in Appendix C: Compound Analogical Design Example. The RaPower case study discussed in Section 5 is also a case of compound analogical design, incorporating regulation feedback from enzymes and color change from the tortoise beetle.

In compound analogy the biological source solution influences the final design outcome. Each analogy brought into the problem changes the conceptualization of the problem itself; modifying the problem model considered for subsequent iterations. In developing the process of compound analogy, however, only the end design was considered; this creates the impression that the analogies are implemented directly to generate a solution to an existing problem aspects. The solution thus generated, it is implied, creates new sub-problems to be solved. However, by considering only the final solution there is necessarily a one-to-one correspondence between analogical source and incorporation of the source into the final solution. Considering the evidence from the RaPower case study, it is reasonable to suggest that many such transfers are made from analogical sources to the problem description; only the final problem description and solution were observed and reported upon in the compound analogy process. Considered across an entire design trajectory, compound analogy as reported, may be a secondary effect of the more routine solution-based problem evolution.

7. Existing Accounts of Problem Evolution

In this section, I will review the past research in design problem evolution, establish the state of the art, and position where my work intersects with the current theories in design problem evolution.

7.1 Defining design as an ill-structured problem

Simon (1973) categorized design as an ill-structured problem. That is, design as a class of problems are inherently under specified and include uncertainty not only with respect to the proper end result, but also with respect to what method(s) might be applied to achieve a result. Even the result itself is subject to uncertainty, in that one may never know whether a design is optimal in an absolute sense. This was similar to Rittel's (Rittel & Webber, 1973) concept of wicked problems. The categorization of design as an ill-structured problem extends as far back as Reitman (1964) who highlights the under-specification of design problems. Dorst (2003) discusses three degrees to which a problem may be ill-defined: (1) some aspects are *determined* by hard requirements, (2) the major part is *underdetermined*, subject to the design choices made by the designer during the process, and (3) some aspects are completely *undetermined* and subject to the style of the designer. In modern design, that design problems are ill-defined is accepted as given; as Cross (2001) put it in his review of 30 years of design studies, "It is widely accepted that design 'problems' can only be regarded as a version of ill-defined problems."

7.2 Two schools of thought on problem structure

There are two core schools of thought on how the structuring of ill-defined design problems is approached. The first school follows the initial work on rational problem solving from Simon (Simon 1973, Newell and Simon 1972). In this view, a problem constitutes a search space, which can be broken down into independent sub-problems, where the sub-problem can be systematical searched until a sufficient solution is identified. The component solutions to sub-problems can then be synthesized into an overall solution. Thus, a problem is first structured, and then a solution is synthesized. Goel and Pirolli (1994) show through a protocol study of design this two-phased approach between problem structuring and solution development, and distinguish it as fundamentally different from other problem-solving activities, which neglect problem structuring almost entirely. Gero (1993) extends the problem-space and search metaphor to suggest that exploration in design is a process which creates new design state spaces or modifies existing state spaces, extending the amount of space which can be explored for design. As Dorst (2003) points out, this rational problem solving approach is representative of the positivist epistemology, suggesting that a problem exists independent of the problem-solver, and can be analyzed and studied objectively, yielding to systematic, scientific processes.

The other school of thought stems from Schön (1983) and what he calls reflective practice. From this viewpoint designers subjectively frame a problem often in conjunction with the generation of one or more possible solutions. While Schön does not explain how such framing occurs, he ties together the inherent subjectivity of the problem as viewed by the designer with the notion that solutions provide a fundamental lever in framing the problem. As he states in later work "problem solving triggers problem setting (Schön 1988)." Dorst (2003) equates this perspective to the phenomenological paradigm in which the construction of reality, in this case the design problem, is inherently subjective. As Dorst and Cross (2001) observe "...designers did not treat design assignments as an objective entity. All designers interpreted the assignment quite differently in awareness of their own design environment, resources and capabilities."

Since the development of these two schools of thought, many case studies, protocol analyses, and performance tests have been conducted usually through one lens or another. Cross (2001) provides a comprehensive summary of 36 studies conducted in the thirty years spanning 1970 – 2000. Relative to problem definition, certain key points of debate have arisen, which can be traced back to the difference in schools of thought on design problems.

7.3 Top-down design

Standard prescriptive methodologies for mechanical engineering and software design (Pahl & Beitz 1984, Roozenberg & Eekels 1995, Dahl, Dykstra, and Hoare 1972, Wirth 1971) suggest a top-down analysis of the problem using function decomposition strategies, for instance. In such a case, in each step of the process detailed design decisions are deferred until the entire problem is sufficiently refined in abstraction. Such processes, while acknowledging the iterative nature of design, suggest that an initial problem formulation can be divorced from solution and analyzed objectively prior to solution instantiation. These methodologies are firmly rooted in the rational problem solving school of thought. Such top-down problem definition strategies are considered as rational, disciplined and well-behaved design (Guindon 1990).

Guindon (1990) observed the practice of a small number of software designers, and showed that rather than applying perfectly top-down strategies, software engineers are more “opportunistic”, and will occasionally be seen to solve a particular sub-problem, prior to returning to the top-down, breadth first problem structuring activity. Likewise, Chadrsekaran (1990) notes that while hierarchical functional decomposition of a problem is an important design task, “in many domains, constraint generation of some sub-problems alternative with partial designs of others, which in turn provide partial constraints for yet other sub-problems.” Ball, et al (1999) counter that, rather than “opportunistic” or “ill-behaved” designers are simply performing a top-down, selective depth-first search which is being invoked to validate the high-level design concept where a designer is unsure. Novices, as one might predict, tend to perform more depth-first problem solving than experts who, being more certain tend to provide breadth-first problem structuring. In 2004, Cross counters with models from Holyoak (1991), Adelson & Solway (1988) and Cross & Clayborn (1998) in which he claims experts do not conform to breadth-first, top-down strategies.

7.4 Solution-orientation

Contrasted with traditional top-down design processes are design studies that show designers often begin with solution conjectures first. Lawson (1979) formalized design as either problem-driven or solution-driven, and characterized the later as being more characteristic of design-based problem solving. The so-called solution-oriented approaches stand in contrast to the top-down, problem-oriented approaches. In such solution-oriented approaches designers quickly conjecture partial solutions to problems, with very little problem structuring or definition occurring prior. Analysis of the proposed solutions can then be used to contextualize and more deeply understand the problem. Hillier, Musgrove, and O’Sullivan (1972) describe the theory of *conjecture-analysis*, which matches observations in architecture and in which early solution conjectures are seen to rapidly reduce the search space by eliminating incongruent alternatives. Darke (1979) expands this theory to *generator-conjecture-analysis* in which a “primary generator”, which can be an idea, or set of coupled ideas, is used to both narrow down the search space and to provide a starting point for the designer. The “primary generator” is imposed on the design problem by the designer. In terms of reflective practice, this primary generator can then be used to frame the problematic design; “set its boundaries, select particular things and relations for attention, and impose on a situation a coherence that guides subsequent moves (Schön, 1988).”

There are however, some potential drawbacks to the solution-oriented approach. Restrepo & Christiaans (2004) show that commitment early, as with solution-oriented approaches, can lead to design fixation. This is not unexpected, as Rowe (1987) observed that “a dominant influence is exerted by initial design ideas on subsequent problem-solving directions...a considerable effort is made to make the initial idea work rather than adopt a fresh point of departure.” This may also be a function of when requirements are produced. Restrepo & Christiaans (2004) show that problem oriented designers produce their requirements throughout the entire

session, whereas solution-oriented designers specified their solution at the beginning of the process. Kruger and Cross (2006) show that for the same problem, some designers employ such solution-oriented approaches, while others use problem-oriented approaches. For their experiment they show that solution-oriented design still tends to produce creative results, but lower overall quality.

Note this traditional notion of solution-orientation is linked directly to the idea of a **conjectured solution**. In this proposal I delineate between two different classes of solutions, and thus two different kinds of solution-based design. The first class, **the conjectured solution**, is the solution class with which most existing theories of design concern themselves. In all instances of design theory mentioned, solution-oriented processes are so-called because they include early solution conjecture. The majority of theories that cite early solution conjectures discuss the relationship between the evolution of the problem, and the conjectured solution. The second class of solution, and the class with which I am most concerned, is the class of **solutions that currently exist**, whether as a solution to the at-hand design problem or a solution that appears unrelated, such as a biological system. My concern is with how such **existing solutions** interact with the design problem. In particular, how **existing solutions** outside of the traditional domain of the design problem – existing biological solutions relative to engineering problems – influence the formulation and evolution of the design problem.

7.5 Problem Decomposition in Design

Problem structuring through decomposition into sub-problems occurs in both solution- and problem-oriented approaches. In prescriptive methods such as Pahl & Beitz (1984), the functional decomposition of a problem into functional sub-problems is an explicitly defined part of the design process. However, some studies show that such decomposition seems to happen even without conscious direction.

Ho (2001) documented the use of both implicit and explicit decomposition of problems. That is, in the observed verbal protocols designers often provided problem decompositions – in this case, a mixed combination of functional and component (form) decompositions – without verbally indicating an intention of creating the decomposition. They note that such implicit decompositions resulted from working forward (depth-first) strategies that were engaged on solving a sub-problem. Then working backward, feedback from the results of this working forward strategy directed the next sub-problem to be considered, again, implicitly.

Liikkan & Pertulla (2009), based on the work of Ho in 2001, applied the implicit and explicit decomposition strategies to their analysis of a group of mechanical engineering students in a controlled experiment setting. They note that approximately 1/3 of the utterances made by designers were problem-oriented, and approximately half of those could be traced to implicit decomposition. They put forth a high-level cognitive model based on their observations that suggests that implicit decomposition occurs during problem interpretation activity, which explicit decomposition occurs during solution generation activity. They note that explicit decomposition, which occurred only twice, had no correlation with the quality of results.

Additionally, they claim that such implicit decomposition is driven from a library of pre-existing decompositions possessed by the designer. The amount and ability to match such relevant decomposition knowledge with an initial problem is dependent on the size of the internal library of decompositions the designer has access to; since novices have smaller internal libraries than experts, they more often producing incomplete or unfit decompositions. This finding coincides with the earlier work of Lloyd & Scott (1994) who posit that domain experience leads to the ability not simply to develop a design, but to structure and decompose a design problem. Restrepo & Christiaans (2004) suggest that while the creation of design requirements is triggered by

prior knowledge, it is also triggered by knowledge acquired during design by interaction with the solution or with external sources of information.

7.6 Time Spent on Problem Structuring

Restrepo & Christiaans (2004) show that information gathered for the purpose of problem structuring, for example about users or the environment, requires additional interpretation and manipulation before it can be used by the designer. This is in contrast to gathering solution information, for example material specifications, the application of which is a known process for the designer.

Christiaans (1992) suggest that “the more time a subject spent in defining and understanding the problem...the better able he/she was to achieve a similar result.” However, from a controlled experiment conducted of 53 engineering students, Atman & Chimka (1999) show that for freshman students design quality is inversely proportional to the amount of time spent on problem definition, whereas it is positively correlated with the amount of time spent in evaluation and decision making. According to Atman & Chimka apparently “some of the freshman students seemed to ‘get stuck’ defining the problem.” On the other hand, for seniors, the amount of time spent in problem scoping highly correlates with the number of constraints their final design satisfied. Seniors also asked for more information during the design process. This suggests that experience plays a significant role productive problem structuring. As Cross (2004) states, “it appears that successful design behavior is based not on extensive problem analysis, but on adequate problem scoping and on a focused or directed approach to gathering problem information and prioritizing criteria.”

7.7 Problem-solution as co-evolution

Maher, Poon and Boulanger (1993) used a slightly different term for the relationship between problem and solution. Rather than classifying the processes as problem- or solution-oriented, they describe the process as a co-evolution. They used the concept of genetic algorithms for a well-defined design problem, to show how such co-evolution could occur computationally. The use of a genetic algorithm required a routine and structured problem definition, limiting the degree to which the technique could generalize, but it served as an interesting proof of concept. Figure 6, adapted from their work shows how starting a problem $P(t)$, a conjectured solution $S(t)$ is generated. This conjectured solution generates new information (shown as a small blue box), which is then transferred to the designers understanding of the problem, generating problem $P(t+1)$. This new problem definition, in turn, is used to generate the next solution iteration $S(t+1)$. The process can iterate until the solution sufficiently meets the requirements in the current problem state.

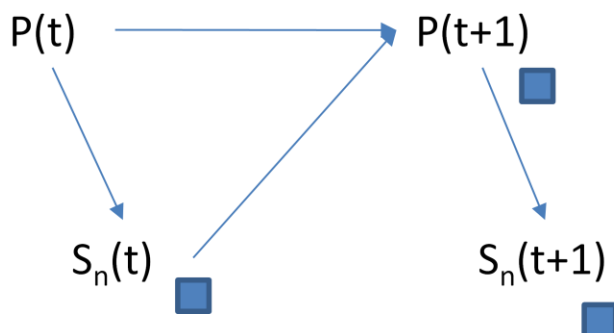


Figure 6. Problem-solution evolution, Maher, Poon and Boulanger (1993)

Subsequently, Dorst & Cross (2001) examine and elaborate on the notion of co-evolution and include in their definition the concept of partial structuring of the solution and problem spaces. In this way, sub-problems could be defined and solved, and the information thus gathered could cycle back through the problem description as partial evolutions of problem and solution. Figure 7 shows the process developed by Dorst & Cross. In this case, the triangle in $P(t+1)$ represents the development of a sub-problem structure, from which a sub-solution $S(t+1)$ is developed. The information from the development of that sub-solution is then cycled back to inform further developments in the problem. Note once again, these are conjectured solutions that are being evaluated in the context of the problem and informing the problem development. The process of sub-problem creation, solution generation, and new problem formulation occurs until all sub-problems are solved sufficiently by the existing array of arranged sub-problems. As described by Dorst and Cross, the process continues until a bridge is built between solution and problem such that the solution to the existing problem is apparent. The metaphor implies the bridge is extended from each side (problem and solution) of the gap, until it makes a solid connection somewhere in the middle.

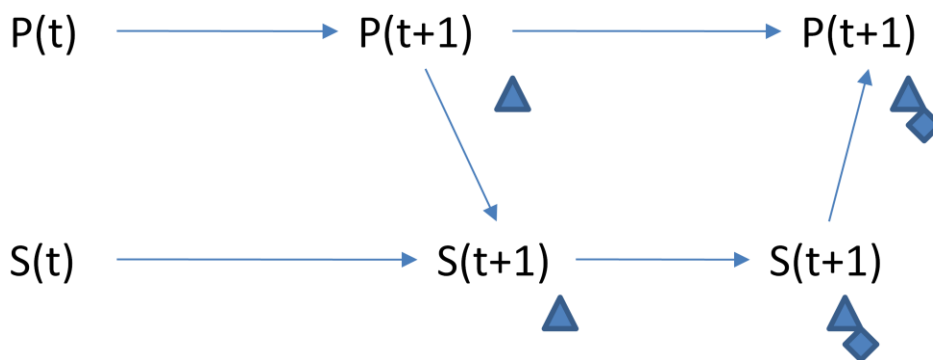


Figure 7. Problem-solution evolution, Dorst & Cross (2001)

While this process reflects the design process, there is a parallel with the work of Nersessian (2009), in which a scientific problem is understood in terms of a simulation. A simulation represents the embodiment of a conjectured solution about the scientific problem at hand. The simulation evolves interactively with the understanding of the scientific problem similar to what is seen in design. While the analogy does not hold perfectly – design is intended to change the world, scientific inquiry to understand it -- there is a striking similarity in the iterative processes seen here.

In the model that I propose, existing solutions also play a role in problem development. Empirically this is supported by the RaPower case, which shows the degree of influence a variety of existing solutions can have on problem formulation. As shown in Figure 8, the modification I make to the process of problem-solution co-evolution is the inclusion of an existing solution that exists at time (0), $S_e(0)$, where the e stands for existing. In subsequent stages the subscript n stands for a conjectured (new) solution. As before, the solution may iterate until solution and problem match occurs. In this case, however, a conjectured solution is not necessary at every

step. At any time a new existing solution can be introduced into the process that lends insight into the current problem, transforming it, and opening up potential new solution paths. This provides a high-level model of what I call solution based problem evolution (SBP-Evo).

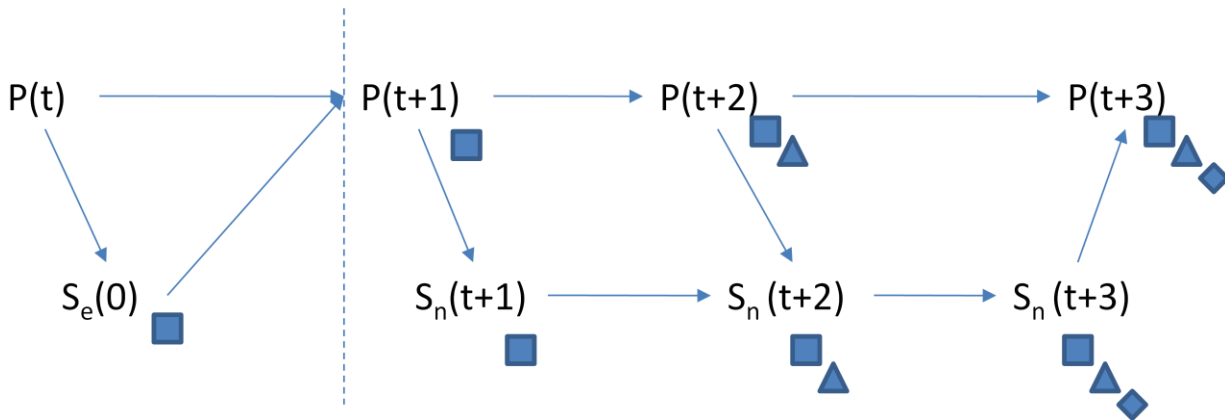


Figure 8. Problem-solution evolution, SBP-Evo

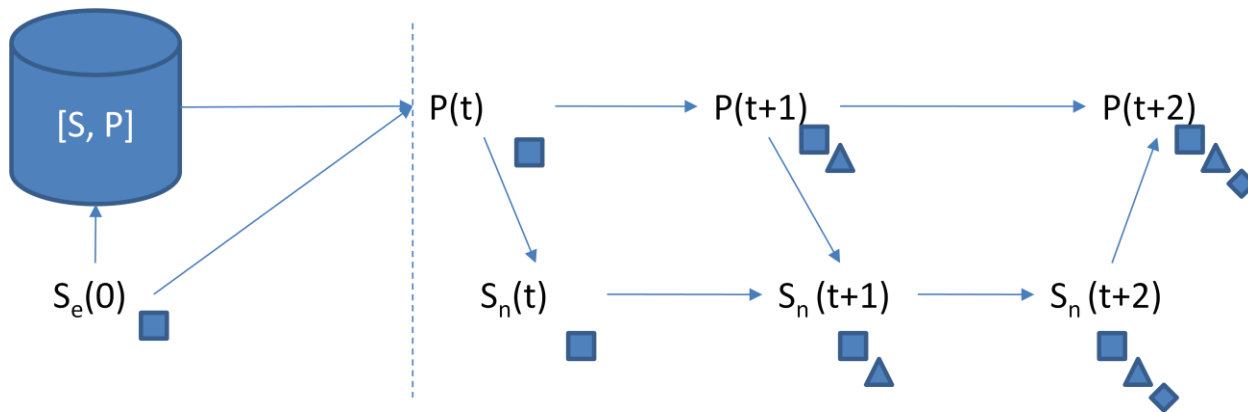


Figure 9. Problem-solution evolution, SBP-Inc

In figure 9, I further extend my own model to account for the so called solution-based inception (SBP-Inc) observations that appear ubiquitous in biologically inspired design. In such an account, the designer begins not with a design problem, but with a particular existing solution (in my context this is always a biological solution), and some internal memory of existing solution-problem pairs, e.g. problems which currently exist and for which there already exist solutions. Through methods explained in more detail in subsequent sections, the solution serves as an index into the memory of problems, and enables the designer to formulate a new problem $P(t)$ from a conjunction of the specifics of the old problem and the existing solution.

8. Existing Accounts and Tool Support for Biologically Inspired Design

In this section I outline the state of the art in biologically inspired design specific theories and tools.

8.1 Process and Cognitive Support for Biologically Inspired Design

Several research groups have evaluated biologically inspired design from a cognitive perspective. Linsey, Wood and Markman (2008), Mak and Shu (2008), Helms, Vattam and Goel (2008, 2009), Helms et al (2008), Vattam, Helms and Goel (2007, 2009) report on cognitive studies of biologically inspired design, while Vincent et al. (2006) proposes a normative theory of biologically inspired design based on the TRIZ theory of innovative design. Wilson and Rosen (2007) provide a process for reverse engineering biological systems to abstract strategies that can be later applied to problems, and Singh et al. (2009) provide a set of strategies for transformation that biological systems employ which might be applied to engineering design. Such strategies may be used to increase efficiency, reduce cost, and increase weight savings.

8.2 Support for Biologically Inspired Design

General design support technology has ranged from interactive design tools that retrieve design drawings (Gross & Do, 1995, Yaner & Goel, 2001) to collaboration across time and space. Following on the growing movement of biologically inspired design, several organizations and research groups are currently pursuing technology agendas for specifically supporting the process of biologically inspired design.

Biomimicry Guild's web portal called AskNature (<http://www.asknature.org/>) provides access to an online functionally-indexed database of research articles in biological sciences. The database is situated in the context of a social networking site enabling designers to better connect with biology researchers.

Chiu & Shu (2007) developed an algorithm that enables engineers to peruse large texts for design-relevant biological systems using functions as search and index terms. Their algorithm uses natural-language analysis, word collocation and frequency analysis, to enable the search and retrieval of relevant biological systems in large text volumes by identifying potential biologically meaningful keywords. Their algorithm was shown to provided for the engineer a set of non-obvious synonyms for function words that may be useful in searching for and retrieving relevant biological systems.

Chakrabarti et al (2005) and Sarkar & Chakrabarti (2008) describe a computational tool (IDEA-INSPIRE) for aiding biologically inspired design, using the SAPPhIRE representation schema to enable functional, behavioral and structural search and referencing of biological source systems. Their work demonstrates, at least in the laboratory context (N = 3), that using their tools with a biologically inspired design process versus non-biologically inspired design process increases the ideation effectiveness of designers by on average 165%. The tool provides the ability to the search of a database of 700 biological entries using the terms in the SAPPhIRE models, and to display a "human understandable" representation of the biological system.

Stone and Wood (2000), developed a functional basis for engineering design, used for modeling systems as flows and functions that transform such flows. The functions are organized into a systematic taxonomy called the function basis; models are called function basis models. Function basis has been extended in several ways for biologically inspired design. First, Shu, et al (2007) show the feasibility of producing function basis models for biologically systems, and then provide a case study demonstrating the usefulness of using function basis models

for analogical transfer between existing biological and technological systems. Importantly, they note the process of analogy may occur at different levels of functional abstraction. Nagel, Stone and McAdams (2010a), further extend the concept of abstraction to include both category and scale abstractions. Cheong, et al. (2011) provide a basis of “biologically meaningful keyword and functional terms” which Nagel, Stone and McAdams (2010b) further extend by providing a “thesaurus” that enables designers and biologists to translate standardized functional basis terminology into biologically equivalent terminology and vice versa, for the earlier developed “meaningful keywords and functional terms.” This further ameliorates the indexing and search problem between the disparate domains.

Vattam, et al (2010) developed a tool, DANE, based on Structure-Behavior-Function (SBF) models and the cognitive models developed in Helms, Vattam, and Goel (2008, 2009). This tool provides designers with both the capability to construct SBF models of biological and technological systems, as well as to search and browse a library of such tools using functional keywords, functional relationships (graph navigation), and keyword search. This tool has been implemented in the context of a biologically inspired design class, where it was useful for generating useful discussions among student teams on the challenges of functional naming, indexing and retrieval, and in the case of one design team, proved useful for helping students structure their design thoughts in terms of abstract functions.

Vattam and Goel (2011) have likewise developed a model based tool for indexing and retrieval of relevant documents associated with biological systems that may be relevant for a design case. Such indexing and retrieval is a common yet challenging task in the classroom environment, where students are required to retrieve many such supporting documents to further their understanding of the biological systems to be transferred.

In the systems developed by Nagel, Stone and McAdams and those developed by Vattam, Wiltgen, Helms and Goel, designers may directly inspect functional models of biological systems. Such systems use these models to not only index and search for relevant biological sources, but also to transfer useful design concepts from biology to the design context. Such models benefit from structural independence, enabling engineers to transfer functional models and then implement with alternative structures more amenable to human manufacture and with performance characteristics specific to the design problem at hand.

These systems above represent the state of the art in technological support for biologically inspired design. The research results for the tools so far developed are focused primarily on indexing, retrieval and transfer of analogies for design; that is, given some target problem, how does one find and transfer the best analogical source system to help solve the design problem. Despite this commonality, one challenge of evaluating these tools is that each uses a different set of design problems, is implemented in a different context, and uses different criteria to evaluate the performance of each system.

Theories developed by Helms, Vattam and Goel, investigate the development of design problems in response to the availability of source systems at the level of a cognitive framework. The work in this proposal builds on those theories. While the current systems outlined above support robust source system modeling, indexing and search, current systems do not address design problems evolution. Design problems are typically represented outside of the supporting system as a part of the test environment, and provided as a design brief or in the form

of a requirements list, or both. Problems are either not represented by the system or are expressed as lightweight and usually fixed models. As seen in the interviews with instructors, however, there is a need for a tool that supports problem evolution and inception in biologically inspired design. One contribution of my thesis is that it can be used to build tools to fulfill this need.

9. Hypotheses: Evaluation

I will propose testing three core hypotheses for my dissertation. The first hypothesis provides a content account for solution-based design in biologically inspired design. The second hypothesis provides an information processing account for solution-based problem evolution, using the content account in hypothesis one. The third hypothesis, which is derivative of the previous two hypotheses, provides an information processing account for solution-based problem inception and solution-based design. This differs from the second hypothesis in that the starting point, instead of a human problem, is a biological solution. The third hypothesis explains the evidence that solution-based design plays a key role in the practice of biologically inspired design.

For each hypothesis offered, I will first provide a plain language description, followed by: (1) motivation, (2) assumptions, (3) method of investigation, (4) Work-to-date, and (5) Schedule of remaining work.

In each case for the method of investigation I will propose two methods for each hypothesis. The first method will be a data analysis component, where validation of the hypothesis will be conducted by comparing the proposed account against case studies of data from prior years BID classes. The case study data has not yet been analyzed in the context of these hypotheses (although in some cases it has been observed), thus providing a test of reliability for the accounts. The second method will be a tool-based component, where a tool that is derived from the account is implemented in a design environment. This implementation will make certain predictions about the design behavior of designers with respect to use of the tool, and will match the expected behavior with design outcomes. The tools will also be measured for frequency of use (usage data) and subjective value (survey and interviews). The table 14 provides a summary of the methods.

Table 14. Summary of Hypothesis Evaluation

	Content Account	SBP-Evo Account	SBP-Inc Account
Case Study	(8) 2008 Case studies	2 cases for each of 2 DPGs from 2009 Case Studies	2 case studies of problem inception from 2009
Tool	2011 problem description tool -Usage -Surveys/Interviews -Consistency and generality measures of P-Schema	2012 problem evolution tool - Usage - Surveys/Interviews - Conceptual trans - DPG Matching	2012 problem inception tool - Usage - Surveys/Interviews - Account of a single instance of inception

10. Content Account for Solution-based Design

10.1 Hypothesis (1)

Designers in biologically inspired design use four different schemas in the process of design problem understanding. These schemas (a) generalize to problems and solutions in biologically inspired design, (b) enabling consistent mapping from observed conceptual elements to a schema element, and (c) provide a sufficient content account to explain the process of interest. The four schemas are:

- (1) Design designer problem goals (DPGs) account for goals relative to understanding the design problem itself, including: conceptual addition, decomposition, partitioning, reorganization, prioritization and constraint relaxation. Each goal thus specified has a beginning state and end state, as well as a set of operators for moving from the beginning to the end state.
- (2) Design problem models in terms of a problem-schema (P-Schema) account for observed data about operational environment, function, artifact specifications, and evaluation criteria.
- (3) Existing solution models in terms of a solution schema (S-Schema) account for observed data about structural specifications, function, behavior relative to an environment, and performance information.
- (4) Solution-problem [S, P] pairs account for the existing solution models in long-term memory that are associated with one or more P-Schema models in long-term memory (i.e. a solution is a solution to one or more problems).

10.2 Motivation

The content account hypothesis describes the underlying information expressed by or inferred from student designers relative to the processes involved in solution-based problem evolution and inception. This represents a small sub-set of the total content used during a complete biologically inspired design process, but it is *sufficient* to explain the *process of interest*. The content account provides the information concepts over which the information processing account operates and includes the conceptual units, their relationships, as well as a memory organization framework.

Building a sufficient content account will (a) help focus students on the important aspects of the information received, (b) enable students to create more discriminating (for design purposes) indexes into memory for new content (especially biological solutions), and (c) enable future retrieval of more salient systems from memory. Furthermore, such a content account (d) creates a shared vocabulary for students from different disciplines, (e) enabling more rapid generation of mutual/common understanding, (f) reducing confusion, and (g) increasing the amount of productive energy spent on generating design solutions.

10.3 Assumptions

I assume that the existing SBF description of solutions is a nearly-sufficient description of biological and human design solutions. SBF models have already been used to model over 50 functions of existing biological systems. Certain limitations for SBF models of specific classes of biological systems are known, including some emergent systems, systems with many similar parts, and homeostatic systems. This work will not address these limitations, and is limited by them.

10.4 Method of Investigation & Evaluation

Three schemas, a P-Schema, an S-Schema, and a DPG Schema will be developed. The P-Schema will be derived using common elements of engineering design and elements derived from classroom observation. The S-Schema will be derived by adding a small amount of new content to the Structure-Behavior-Function (SBF) schema already used to represent systems in biologically inspired design. The DPG-Schema will be developed by abstracting concepts from problem transformations recorded in class. Since the DPG-Schema have start and end states that are P-Schema, the DPG-Schema will be described using abstract patterns of P-Schema.

Validation of the content account requires meeting three criteria:

- (1) The schemas developed for the content account must be general enough to describe the observed data,
- (2) The schemas must be specific and detailed enough that mapping between the schemas and observed data is consistent, and
- (3) The schemas must provide explanatory power in the context of a processing account.

Two methods of investigation are used to validate the schemas developed in support of this hypothesis:

10.4.1 Method 1: Data Analysis

I will use a set of data collected from 8 class projects in 2008 that provide detailed problem and solution descriptions, as well as patterns of transformation over time which represent the DPGs. This data includes problem descriptions in the form of homework assignments gathered at four different points during the class. These homework assignments often also include solution descriptions, both biological “source” solutions, and new proposed solutions. The data from the 2008 instances of biologically inspired design will be mapped into the schemas (P-schema, S-schema, and DPG-schema).

10.4.2 Method 1 Evaluation

The degree to which the schemas are capable of representing all of the data present will be measured to *support generality* across design projects. Schemas will be adjusted, if necessary, until they generalize across roughly 90% of relevant design data.

The same mapping will be evaluated using multiple coders for some sub-set of the data. The degree to which the coding agrees will then be measured to *support consistency*. Schemas and definitions will be adjusted until coding consistency reaches generally acceptable levels for the method employed.

This method creates an opportunity for bias – that schemas will become too tailored to the highly situated 2008 data set. Method 2 for this hypothesis, as well as work in hypothesis 2 and hypothesis 3 will be used as a check against this bias, using data sets from 2009 and 2011.

10.4.3 Method 2: Tool Deployment

A tool will be developed and deployed in the Fall 2011 biologically inspired design class to further validate the p-schema. The tool will provide student designers with the framework and definitions of the p-schema. Students will be asked throughout their design projects, as part of assignments already embedded in the class, to provide problem descriptions, in the form of p-schema descriptions.

10.4.4 Method 2 Evaluation

Student descriptions will be evaluated for consistency of use of the p-schema. A subjective determination will be made to determine whether content designated in the student p-schema, for instance *function*, matches the p-schema definition for that conceptual designation. The degree to which students map concepts correctly between their design problem and the p-schema will determine the *consistency* of the p-schema *as used prescriptively*; the degree to which all concepts can be mapped to elements of the p-schema will determine *generality* of the p-schema *as used prescriptively*. In the second case, since it is possible that some conceptual elements may be dropped because they are not able to be represented in the p-schema, the tool will provide a catch-all category. This will enable students to express problem concepts not represented in the p-schema. Additional survey data (see next paragraph) will be used to identify the presence of design problem concepts that the p-schema could not represent.

A pre-class survey will be conducted eliciting the student's ideas with respect to what are important concepts for describing and understanding design problems. This will provide a base-line reference. The students will be taught how to use the p-schema tool. A survey will be conducted after course to determine (a) the perceived value of the tool relative to the task of design in the biologically inspired design class, (b) changes in thinking about design problem formulation relative to the base-line survey, and (c) any specific examples of failure of the p-schema tool for representing problems.

10.5 Work-to-date

A complete problem schema (P-Schema) has been developed (see Appendix E: Problem Schema), as well as small extensions to the existing SBF Schema, which represents the solution schema (S-Schema). The extensions include performance indicators in terms of benefits and deficiencies for the particular solution schema with respect to a given problem. These performance indicators are first class objects in the schema.

The fourth schema, the [S, P] pairs, exist as P-Schema and S-Schema elements each of which is already defined. Currently these pairs are linked through a high-level "pair" object that provides a pointer to one [S] and one [P] model. However, other conceptual links may (and almost certainly do) exist at lower levels across the [S] and [P] models, such as close functional relationships; these lower level conceptual links may provides a "strength of pairing" weighting that may be important in the information processing account.

Method 1: Two of eight of the projects from the 2008 class have been evaluated in detail, with a generalization of 92%. Cross-coding validation was not done for these cases. All eight final projects were evaluated by a master's student using the schema to generate indexes for a technology platform. Analysis of that exercise shows that additional degrees of specification are required for one of the major categories, "artifact specification", and some limiting of the scope of "criteria" needs to occur.

Method 2: The tool and survey instruments have been designed for the class, and are ready for deployment in the 2011 Fall class. IRB is complete for all classroom experiments through Fall 2012.

10.6 Remaining work

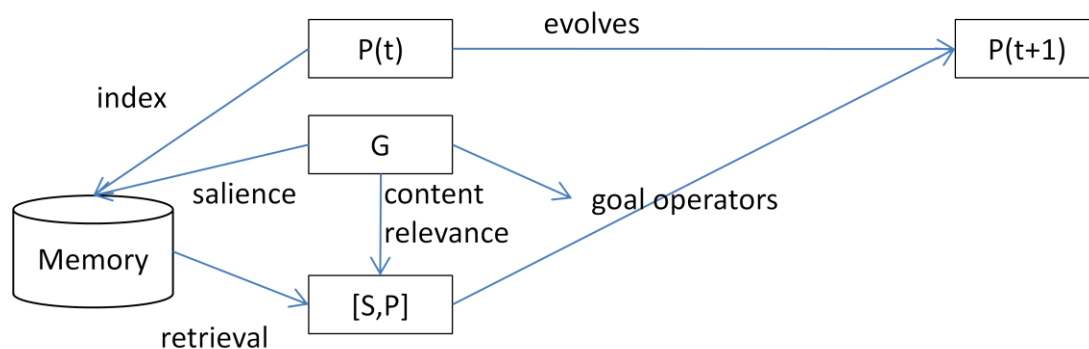
Method 1 evaluation of remaining 2008 cases will take place concurrent with the Fall 2011 experiment.

Method 2 evaluation of survey data and data collected on problem descriptions using the p-schema will take place in December 2011, after the class is complete. Preliminary evaluation may occur during the class as data is received.

11. Information Processing Account of Solution-Based Problem Evolution

11.1 Hypothesis (2)

Given a starting problem $P(t)$, and a designer problem goal G , one method of problem elaboration is by indexing, accessing and transferring problem concepts from a solution-problem $[S,P]$ pair in memory. The particular designer problem goal G , determines the indexing salience, transfer operators and content transferred. The process yields a new problem description $P(t+1)$.



11.2 Motivation

Analysis of the RaPower 2008 design case provides an example of the strong influence that existing solutions exert on student conceptualization of their design problem. For example in the final design *more than 75% of the functions and almost 50% of the operational environment concepts* considered in the final problem description originated from earlier descriptions of *existing solutions*. Moreover, we know from firsthand observation and from instructor requirements that student designs find problem specification difficult in the context of open-ended problems in the biologically inspired design class. It is not currently known why or what aspect of problem evolution students find most challenging in the context of biologically inspired design.

Empirically, in the RaPower case study 10% of concepts originally associated with a given problem followed through to the final design. Thus throughout the course of a complete design cycle, students discarded 90% of the concepts associated with a problem along with most of the work on solutions related to the discarded concepts. While the generation of concepts that are later discarded may be a necessary product of the creative design process, optimizing conceptual turnover and solution generation activity – for instance, by generating and discarding a large number of concepts earlier, rather than later -- may yield a deeper focus on solution development for a longer period. From a purely theoretical standpoint, for two design processes that result in the same end problem concepts, the design process with less conceptual turnover will be more efficient.

Current cognitive theories of problem evolution focus on problem evolution at some time after a *new solution* (or new solution analogue) has been proposed and evaluated. In this case, the evolution of the problem is usually driven from some deficiency in the proposed solution that is detected during evaluation. This deficiency generates new problem specifications that must also be met as part of generating a solution to the problem. Our evidence suggests that in addition to post-evaluation problem evolution, evolution also occurs also as a

result of the *consideration of existing solutions*. In the RaPower case study in biologically inspired design for certain concept classes, in particular for functions, the majority of additional problem concepts were generated by considering existing solutions.

Since problem evolution shapes the ultimate solution design, and given the data from our case study, providing a specific information processing account of solution-based problem evolution will enable a deeper understanding of how students access, evaluate and transfer information from existing solutions to their problem model.

With a more specific model of the activity of solution-based problem evolution, it may be possible to focus students on more productive conceptual transfer to their problem model, and prevent wasteful conceptual turnover. A rich information processing account will provide a framework for inquiry to understand why and when students struggle with the process of problem evolution, and should lead to better teaching strategies. Finally, an information processing account enables the development of tools to support solution-based problem evolution in the context of biologically inspired design.

11.3 Assumptions

I assume that the conceptualization of a design problem, that is the set of associated concepts and their relationship used to describe a problem, influences the solution generated for that design problem.

11.4 Method of Investigation & Evaluation

Using the schemas (P-schema, S-schema, DPG, and [S,P] pairs) developed previously, I will develop an information processing account for solution-based problem evolution (SBP-Evo) that provides an explanation for how, given the design goal for problem elaboration, a library of existing solutions can be used to further enhance a problem description.

I will develop sufficient information processing accounts for two of the DPGs (see Appendix F: Designer Problem Goals) described in the content account. This includes complete descriptions of the DPG initial and end patterns, saliency weighting for the recall step for each DPG, and the set of information transfer and transformation processes necessary to change $P(t)$ to $P(t+1)$.

Validation of an information processing account can occur in at least three ways:

- a) The first is to build an “intelligent” system that uses the information processing account to perform the process described in the account in an automated way. Because knowledge engineering demands are usually high, it is often the case that these end up as “toy-world” proofs-of-concept for the account.
- b) The second is to use the information processing account to explain a set of existing data observed in the real world. One can explain existing *in-situ* or *in-vivo* accounts in this way, or can create controlled *in-vitro* environments to limit some of the potentially confounding variables found in the other environments. This provides a proof-of-concept grounded in real-world data.
- c) The third is to use the information processing account under different conditions to make and test predictions of the activity which the account describes. The information processing account can be embodied as a tool used to support the activity the information account purports to describe.

Controlling the conditions under which the tool is used provides a set of outcomes that can be compared to the expected outcomes predicted by the account under those conditions.

I will evaluate the information processing account using the last two methods.

11.4.1 Method 1: Data Analysis

I will use the SBP-Evo account to explain a set of observed data. In particular, I will use the SBP-Evo account to describe instances of problem evolution from design cases from 2009. Since this process necessarily includes mapping case study data to the schemas developed previously, using cases from 2009 will further validate the content account hypothesis, without the potential 2008 case study bias. I will take two instances from the 2009 case studies for each of the two DPGs developed in the information processing account, and provide a detailed explanation for how student designers evolved their problems using the information processing account.

11.4.2 Method 1: Evaluation

For this method of validation it is necessary to demonstrate that existing data collected independent of the formation of the account, can be explained using the information processing account. There are at least three tractable methods for evaluating the mapping of observable data to an information processing account which by its nature explains some amount of necessarily “unobservable” data (e.g. the memory and information processing that occurs in the mind:

- (1) The input and output of the designer(s) must match the content account that the information processing account is based on.
- (2) The output of the designer(s) must match at least one output condition the information processing account could predict, based on the input condition.
- (3) One must be able to provide a proposed set of unobserved data (data in memory) that would be capable of explaining the observed transformation from input to output using the transformation rules and heuristics provided for in the information processing account.

Each data point (two points for each of two DPGs, four total) will provide an input and output description from the 2009 case studies. To meet the three criteria listed above, first the input and output will be mapped to the content account. Second, the input and output pairs will be matched to the respective input and output pairs that are part of the definition of the DPG (see Appendix F). And finally I will describe one or more underlying models in memory and the transformations on the observed input, using those models and the transformation rules and heuristics in the SBP-Evo account, to generate the observed output. This provides a data grounded validation for the SBP-Evo information processing account.

11.4.3 Method 2: Tool-Based Experiment

I will implement a technology-based problem evolution tool for use by BID student designers based on the SBP-Evo information processing account. The tool will enable students to provide a problem description in p-schema form, and access a library of solutions in external memory (a large number of S-schema solution models are already available in several databases). Solution databases of S-schema models will be augmented by their corresponding problem pairs such that the predicted [S, P] pair schema will be embodied in these external databases.

Access to [S, P] pairs will be logged as well as changes in problem descriptions that occur over time. Changes to problem descriptions will be matched against predicted changes using the DPGs and the observed [S, P] pairs. By monitoring the accessed [S, P] pairs in memory and the starting state of the problem description, the information processing account will provide predictions about the kinds of problem transformations and the content of those transformations that will occur during problem evolution cycles of the design episode.

11.4.4 Method 2: Evaluation

Evaluation will be based on (1) the effectiveness of the tool as determined by student use, (2) the effectiveness of the tool as determined by instructor evaluation, (3) the effectiveness of the tool to index and access the solutions databases, and (4) the effectiveness of the SBP-Evo model tool influence problem evolution.

- (1) Evaluation of the effectiveness of the tool based on student use will be conducted using (1) usage data, and (2) survey data. The usage data will report the number and kind of uses for the tool over the course of the design episode. In particular the tool will record statistics with respect to (a) problem evolution activity and (b) solution database access activity. The effectiveness of the different aspects of the tool will be measured in terms of total use (activity), and relative use internally (activity of one tool component with respect to the other) and relative use externally (compared to the activity of the database tools that were used in their standalone forms). A survey will be provided to student users after their experience with the tool to determine the perceived usefulness of the tool to the students.
- (2) Evaluation of the effectiveness of the tool based on instructor evaluation will be conducted using instructor interviews. Instructors will be asked to compare the performance of students that have used the tool, versus students that have not (for instance in previous years BID classes).
- (3) Evaluation of the effectiveness of the tool to index the solutions databases will observe absolute measurement of accesses to the database, and correlations between problem descriptions and solutions accessed. The theory predicts that certain problem descriptions will serve as more dominant indexes to the database than others, for instance functional indexing will account for more than operational environment indexes, and artifact specification indexes will account for the least.
- (4) In addition to accessing the tool, observations will be made for conceptual transfer from the [S, P] pair to the problem model. This will take the form of one or more concepts from the [S, P] pair occurring in the problem-model after the [S, P] pair has been accessed. The amount and types of concepts transferred will serve as one metric. The before and after states of the transfer will also be matched to states specified in the DPGs. To the extent this can be used to identify DPGs, these will also be used to classify the types of conceptual transfer that occur. States that do not correspond to existing DPGs, to the extent that they form new patterns of transfer, may indicate new DPGs, or different strategies for addressing a particular DPG.

11.5 Work Completed

This draft of the SBP-Evo account is contingent on the content account, and is thus subject to change as results from the Fall 2011 experiment are made available.

Databases of existing S-schema models exist in three separate tool contexts: DANE, Biologue, and a BID case indexing tool currently in development. These all use the SBF modeling language which will require only minor modification to serve as the s-schema library.

Based on the input from the 2011 experiment an electronic tool for describing a problem based on the p-schema must be developed. An early web-based prototype of this tool, called Blue*Spark, was begun in April 2009 but development was discontinued. The database and infrastructure from this tool can be reused to build the required tool.

The tool developed to test the p-schema framework itself for hypothesis 1 will establish a baseline interface and will serve as a first test of the required functionality for the tool.

11.6 Remaining Work

Joining the existing database developed in 2009 with the interface developed in 2011 will provide the necessary tool for complete problem description and tracking.

A unified interface to the existing solution databases will also be deployed. This interface will enable access to the existing solutions, searchable by relevant [S, P] pair fields. A new presentation layer will be developed for the [S, P] pair that includes relevant information from both S-schema and P-schema models (current tools provide on S-schema information). The unified interface will enable user tracking and access to different models.

All S-schema models need to be augmented with incremental S-schema information, as well as a relevant P-schema model describing the problem solved by the solution. The P-schema model will necessarily be a lightweight model.

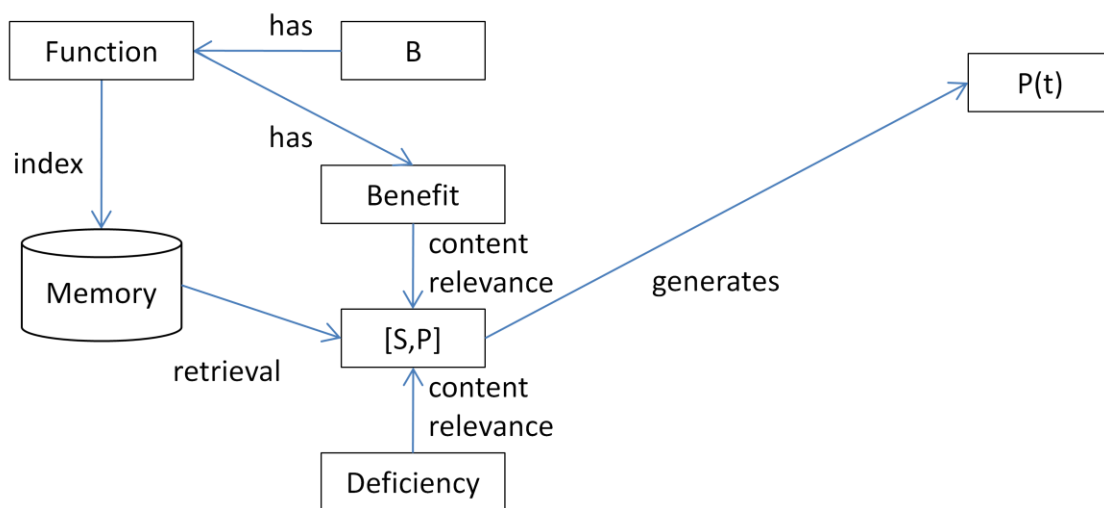
Once built, the tool must be deployed in an *in situ* design context. The most obvious context is the Fall 2012 BID class, although this need not necessarily be the case. Any instance of *in situ* BID student design will suffice. The Center for Biologically Inspired Design often has many such student projects underway at any given time.

Data evaluation can take place during and immediately following the class, in Fall 2012 or earlier for a project not based on class.

12. Information Processing Account of Solution-Based Problem Inception

12.1 Hypothesis (3)

Given a starting biological solution (B), a solution-problem [S,P] pair can be found in memory with either a structural, functional or environmental similarity (functional shown in diagram) matching an index provided by the biological solution. If the performance of a portion of the solution in the [S, P] pair is inferior in some way to the performance of an element of the biological solution, a new problem (P) can be generated from the problem specification provided in the [S, P] pair, and from the any relative deficiency between matching elements in the biological solution (B) and the existing solution (S).



12.2 Motivation

One characteristic of biologically inspired design is the appearance of a design problem in response to the observation of a biological solution. That is, when a student designer learns about a particular biological solution, for example the impact resistant characteristics of abalone nacre, the student designer generates a new problem for which the biological system (e.g. the abalone shell) will provide a good solution. This solution-based design pattern accounts for almost 50% of the projects observed in the 2006 and 2007 biologically inspired design classes. Many of the most successful commercial designs – Velcro and Lotusan Paint are two of the more prominent – likewise appear to follow this design pattern. In an analysis of 60 biologically inspired designs [Helms, Vattam, & Goel, 2007], almost 80% of those that are either currently in the market or developed as marketable prototypes were solution driven.

An information processing account for this process provides practitioners and educators with a model for how and why this process occurs, and may lead to an account of why it is successful. Furthermore, almost all work in biologically inspired design is concerned with the process of finding relevant biological solutions to pre-defined

problems – much of it situated in the context of search and retrieval of relevant biological systems for a given problem. This account, which addresses a majority of real-world cases, turns the traditional problem-driven process inside out, providing researchers with an alternative approach to biologically inspired design and subsequent research. Already we see our work on solution-driven biologically playing a prominent role in shaping at least one major line of research outside of our lab (Shu, 2011).

12.3 Assumptions

I assume the SBF representation is a sufficient solution representation. I further assume that a designer starts with a long-term memory of solution-problem pairs.

12.4 Method of Investigation and Evaluation

Using the schemas (P-schema, S-schema, DPG, and [S,P] pairs) developed previously, I will develop an information processing account for solution-based problem inception (SBP-Inc) that provides an explanation for how, given an initial starting biological solution, B, and a library of existing [S, P] pairs, a new problem P(t) can be generated to which B provides a partial solution. This P(t) can then serve as a starting point for problem evolution, per Hypothesis 2.

I will evaluate the information processing account using the same general methods as in the previous hypothesis.

12.4.1 Method 1: Data Analysis

I will use the SBP-Inc account to explain a set of observed data. In particular, I will use the SBP-Inc account to describe instances of problem inception from design cases from 2009. Once again, since this process necessarily includes mapping case study data to the schemas developed previously, using cases from 2009 will further validate the content account hypothesis, without the potential 2008 case study bias. I will take two instances from the 2009 case studies and provide a detailed explanation for how student designers generated their problems using the information processing account.

12.4.2 Method 1: Evaluation

For this method of validation it is necessary to demonstrate that existing data collected independent of the formation of the account can be explained using the account. The same methods apply for evaluating the mapping of observable data to an information processing account as in the previous hypothesis.

For two examples of solution-based problem inception, I will use input and output descriptions from the 2009 case studies. To meet the three criteria listed above, first the input and output will be mapped to the content account. The second criterion is met by satisfying the condition that the initial input (a solution B), is at least a partial solution to the output (a new problem P). Finally I will describe one or more underlying models in memory and the transformations on the observed input, using those models and the transformation rules and heuristics in the SBP-Inc account, to generate the observed output. This provides a data grounded validation for the SBP-Inc information processing account.

12.4.3 Method 2: Tool-Based Experiment

I will implement a technology-based problem inception tool for use by BID student designers based on the SBP-Inc information processing account. The student designers will either be enrolled in an existing biologically

inspired design class, or engaged in a mentored biologically inspired design project. Several such mentored projects are often ongoing through the Center for Biologically Inspired Design. The tool will enable students to specify a biological solution in S-schema form. From that specification student designers will be able to access the existing database of [S, P] pairs, and select a relevant [S, P] pair if one exists. Given their own biological solution, and a relevant an [S, P] pair retrieved from that database, or from their own set of solution problem pairs, the student designers will be able to specify a new problem using the problem description tool from Hypothesis 2.

The biological solution and access to the database [S, P] pairs will be logged as well as the any new problem description. By monitoring the accessed [S, P] pairs in the database and the starting solution, the information processing account will provide predictions about the kinds of problems generated and the content of problems generated.

12.4.4 Method 2: Evaluation

Evaluation will be based on (1) the effectiveness of the tool as determined by student use, (2) the effectiveness of the tool as determined by instructor/mentor evaluation, (3) the effectiveness of the tool to find a relevant [S, P] pair in the database, and (4) the effectiveness of the SBP-Inc tool to influence problem inception.

This evaluation method assumes that a relevant [S, P] pair can be found either in the database, OR in student memory during an open-ended *in situ* design project. Since the number of problems which may be solved by a particular solution is potentially very large, and since there are countless many biological sources from which students may choose there is no guarantee that a finite database of a hundred or so [S, P] pairs will contain a relevant match. In this case, I am hoping that student bias toward the strengths and preferences of their instructors will guide at least a few students toward a set of biological solutions similar to those addressed in previous years. If student do use one or more systems from the database, evaluation **Protocol I** will be used. Students may, however, select some solution-problem pair from their own memory, in which case evaluation **Protocol II** will need to be used.

Elements (1) and (2) of each protocol will apply to all participants.

Elements (3) and (4) of each protocol will apply to a single design project, selected according to the following criteria:

- a. A design project that uses a database solution-problem pair will be used in preference to a project that accesses a solution-problem pair from student or other external memory source.
- b. A design project with readily apparent conceptual transfer will be used in preference to a project without readily apparent conceptual transfer.
- c. The design project with a greater amount of conceptual problem elaboration, as judged by the formula (number of concepts + number of conceptual relationships) will be used in preference to a design project with less conceptual elaboration. This uses the P-schema model of problem concepts and relationships.

12.4.4.1 Evaluation Protocol I

- (1) Evaluation of the effectiveness of the tool based on student use will be conducted using (a) usage data, and (b) survey data. The usage data will report the number and kind of uses for the tool over the course of the design episode. In particular the tool will record statistics with respect to (i) problem activity and (ii) solution database access activity. The effectiveness of the different aspects of the tool will be measured in terms of total use (activity), and relative use internally (activity of one tool component with respect to the other) and relative use externally (compared to the activity of the database tools that were used in their standalone forms). A survey will be provided to student users after their experience with the tool to determine the perceived usefulness of the tool to the students.
- (2) Evaluation of the effectiveness of the tool based on instructor/mentor evaluation will be conducted using instructor interviews. Instructors will be asked to compare the performance of students that have used the tool, versus students that have not (for instance in previous years BID classes).
- (3) For one design project, evaluation of the effectiveness of the tool to index the solutions databases will observe absolute measurement of accesses to the database, and correlations between problem descriptions and solutions accessed. The theory predicts that certain problem description concepts will serve as more dominant indexes to the database than others, for instance functional indexing will account for more than operational environment indexes, and artifact specification indexes will account for the least.
- (4) In addition to accessing the tool, for one design project, observations will be made for conceptual transfer from the [S, P] pair to the problem model. This will take the form of one or more concepts from the [S, P] pair occurring in the problem-model after the [S, P] pair has been accessed. The model makes predictions about what kind of conceptual information will be transferred, and what will not; again in order: functions, operational environment, and artifact specifications. Furthermore, the theory predicts that a specific benefit or deficiency, related to one of the other transferred concepts, will provide the focal point for the transfer.

12.4.4.2 Evaluation Protocol II

- (1) Student usage and survey data, per protocol I.
- (2) Instructor/mentor survey data, per protocol I.
- (3) Index use data, per protocol I.
- (4) Since the students are accessing a problem from memory, the models for the existing theorized solution problem pair will not be available. Evaluation will be conducted through (a) collection of any external source information used in the process (e.g. papers, web references, textbooks, etc.) and (b) open-ended interviews of the individual team members to elaborate on the design trajectory. In particular the interviews will solicit information about (i) the origination of the selected problem; (ii) when during the design process was the problem selected; (iii) why that problem was selected; (iv) what the most relevant aspects of that problem were; (v) what aspects of the problem were dropped. The information so gathered will be mapped to a P-schema, and matched against predictions made by the information processing account, as per protocol I.

12.5 Work Completed

The draft of the SBP-Inc account is contingent on the content account, and is thus subject to change as results from the Fall 2011 experiment are made available.

Initial cases of solution-based design from 2007 and 2008 have been informally mapped to the information processing account. Through this informal mapping, conducted summer 2010, formal mapping for two cases is both tractable and should pose minimal difficulty.

As mentioned in the previous hypothesis, databases of existing S-schema models are already in place.

The first iteration of the P-schema description tool is already complete, per hypothesis 1. This is the primary tool that students will use to provide a description of their new problem. A second iteration will follow pending results from the Fall 2011 study.

12.6 Remaining Work

All of the technical work from hypothesis 2 will be reused for expression of problems as P-schema and for the presentation of the P-schema and S-schema pairs. No additional end user functionality is anticipated for the evaluation of this experiment.

Again, the tool must be deployed in an *in situ* design context, with the most obvious being the Fall 2012 BID class, although this need not necessarily be the case. Any of the instances of *in situ* BID student design that was have access to should be sufficient. The Center for Biologically Inspired Design often has many such student projects underway at any given time, providing some scheduling flexibility.

Deployment will require some designer training and a specific solution-based design process. It is process, not the technology, which is the significant difference between this and the experiment conducted for the previous hypothesis. If conducted in class, this process is already instituted through the mandated solution-based design project.

Data evaluation can take place during and immediately following the Fall 2012 class or earlier for a project that takes place outside of class.

13. Timeline for completion

Theoretical Work		
Work Description	Details	Completion Timeframe
S-schema defined	Augmented SBF with benefit/deficiency [b/d] information	Complete.
P-schema defined	Defined in Blue*Spark	Complete, subject to Fall 2011 test and revision
SBP-Inc defined	Partially completed SBP-Inc information processing account	Subject to Fall 2011 test and revision of schemas; complete Spring 2012
SBP-Evo defined	Partially completed SBP-Evo information processing account	Subject to Fall 2011 test and revision of schemas; complete Spring 2012

Technical Work		
Work Description	Details	Completion Timeframe
Database schema for [S-schema models]	Partially Implemented in DANE and Biologue; requires augmentation with [b/d] information.	Complete.
Database schema for [P-schema models]	Implemented in Blue*Spark	Complete, subject to Fall 2011 test and revision.
Database populated for [S-schema models]	Base SBF models are complete, required [b/d] information to be added. Minor.	Fall 2011
Database populated for [P-schema models]	The majority of the database work, requires a problem description for each [S] model in DANE and Biologue	Fall 2011
Framework for student use of P-schema models	Framework for input of P-Schema models	Complete, subject to Fall 2011 test and revision
P-Schema model creation process	Process for students to create of P-Schema models	Complete, subject to Fall 2011 test and revision.
Framework for student access to S-schema models	Single GUI for viewing S-Schema models from different databases	Spring 2012
S-Schema model creation process	Process for the creation of S-Schema models	Complete, subject to Fall 2011 test and revision
Integrated [S,P] viewing	View both S and P models as a pair from multiple databases.	Summer 2012
[S,P] search	Search for S-Schema, P-Schema or [S,P] pairs	Summer 2012

Experimental work		
Work Description	Details	Completion Timeframe
Test P-Schema Data	Map 2008 projects to P-schema	Fall 2011
Test P-Schema Tool and Process	Implement P-schema creation tool in classroom, Fall 2011	Fall 2011
Analyze P-Schema Results	Analyze the results from data and tool deployment	Spring 2012
Test SBP-Evo Data	Map 4 problem evolution instances from 2009 projects to SBP-Evo	Summer 2012
Test SBP-Evo Tool	Implement SBP-Evo tool in classroom or on alternative project.	Summer-Fall 2012
Analyze SBP-Evo Results	Analyze the results from data and tool deployment	Fall 2012
Test SBP-Inc Data	Map 2 problem inception instances from 2009 project to SBF-Inc	Summer 2012
Test SBP-Inc Tool and Process	Implement SBP-Inc tool in classroom or on alternative project.	Summer-Fall 2012
Analyze SBP-Inc Results	Analyze the results from data and tool deployment	Fall 2012

14. References

- Adelson, B., & Solway, E. (1988) A Model of Software Design, in Chi, M, R Glaser and M Farr (eds.), *The Nature of Expertise*, Erlbaum, Hillsdale, NJ, USA.
- Atman, C., & Chimka, J. (1999) A Comparison of Freshman and Senior Engineering Design Processes. *Design Studies* 20(2):131-152.
- Ball, L., Evans, J., Dennis, I., Ormerod, T., (1997) Problem-Solving Strategies and Expertise in *Engineering Design. Thinking and Reasoning* 3(4):247-270.
- Bar-Cohen, Y. (2011) *Biomimetics: Nature-Based Innovation* in Press. CRC Press.
- Benyus, J. (1997) *Biomimicry: Innovation Inspired by Nature*. New York: William Morrow.
- Bhatta, S., & Goel, A. (1996) Model-Based Indexing and Index Learning in Engineering Design. *Engineering Applications of AI, Special issue on Machine Learning in Engineering*, 9(6):601-610.
- Bonser, R., & Vincent, J. (2007) Technology Trajectories, Innovation, and the Growth of Biomimetics. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, pp. 1177-1180
- Chakrabarti, A., Sarkar, P., Leelavathamma, B., & Nataraju, B. (2005) A Functional Representation for Aiding Biomimetic and Artificial Inspiration of New Ideas. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 19:113-132.
- Chandrasekaran, B., (1990) Design Problem Solving: A Task Analysis, *AI Magazine*, 11(4), pp.59-71.
- Cheong, H., Chiu, I., Shu, L., Stone, R., McAdams, D. (2011) Biologically Meaningful Keywords for Functional Terms of the Functional Basis. *Journal of Mechanical Design*, 133.
- Chiu, I., and Shu, L. H., (2007) Biomimetic Design Through Natural Language Analysis to Facilitate Cross-domain Information Retrieval. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, **21**:45–59.
- Christiaans, H (1992) *Creativity in design* PhD Thesis, Delft University of Technology, Delft, The Netherlands.
- Cross, N. (2001) Design Cognition: Results of Protocol and Other Empirical Studies of Design Activity. In C. M. Eastman, W. M. McCracken, & W. C. Newstetter (Eds.), *Design Knowing and Learning: Cognition in Design Education*. (pp. 79-103). Amsterdam: Elsevier.
- Cross, N. (2004) Expertise in Design: an Overview. *Design Studies* 25(5):427-441.
- Cross, N & Clayborn Cross, A. (1998) Expertise in Engineering Design, *Research in Engineering Design* 10:141-149.
- Dahl, O.J., Dykstra, E.W., & Hoare, C.A.R. (1972). *Structured programming*. New York: Academic Press.
- Darke, J. (1979) The Primary Generator and the Design Process. *Design Studies*, 1(1):36-44.
- Dorst, K. (2003) The Problem of Design Problems. In N. Cross & E. Edmonds (Eds.), *Expertise in Design* (pp. 135-147). Sydney: Creativity and Cognition Studio Press.
- Dorst, K., & Cross, N. (2001) Creativity in the Design Process: Co-evolution of Problem-Solution. *Design Studies*, 22(5):425-437.
- Gross, M., Do, E., (1995) Diagram Query and Image Retrieval in Design. In *2nd IEEE International Conference on Image Processing*, Washington, D. C., IEEE Computer Society Press.

- Goel, V., & Pirolli, P. (1992) The Structure of Design Problem Spaces. *Cognitive Science*, 16: 395-429. Gero, J.S. (1993). Towards a Model of Exploration in Computer-Aided Design, In J.S. Gero and N. Tyugu (Eds.) *Formal Design Methods for Computer-Aided Design*, North-Holland, Amsterdam.
- Guindon, R. (1990) Designing the Design Process: Exploiting Opportunistic Thoughts. *Human Computer Interaction*, 5:305-344.
- Helms, M., Vattam, S., Goel, A. (2008) Compound Analogical Design, or How to Make a Surfboard Disappear. In B.C. Love, K. McRae, & V.M. Sloutsky (Eds.) *Proceedings of the 30th Annual Conference of the Cognitive Science Society*, Washington D.C.: Cognitive Science Society
- Helms, M., Vattam, S., & Goel, A. (2009) Biologically Inspired Design: Products and Processes. *Design Studies*, 30(5):606-622.
- Helms M, Vattam S, Goel A (2010) The Effects of Functional Modeling on Understanding Complex Biological Systems. *Proc. ASME Int. Design Engineering Technical Conf.*, Montreal, Canada.
- Helms, M., Vattam, S. Goel, A., Yen, J., Weissburg, M. (2008) Problem-Driven and Solution-Based Design: Twin Processes of Biologically Inspired Design. In *Silicon + Skin: Biological Processes and Computation, Proceedings of the 28th Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA)*, Minneapolis, MN.
- Hillier, W., Musgrove, J., & O'Sullivan, P., (1972) Knowledge and Design, in Mitchell, W J (Ed.) *Environmental Design Research and Practice*, University of California, US.
- Ho, C-H., (2001) Some Phenomena of Problem Decomposition Strategy for Design Thinking: Differences Between Novices and Experts, *Design Studies*, 22(1): 27-45.
- Holyoak, K. J. (1991) Symbolic Connectionism: Toward Third-generation Theories of Expertise, in Ericsson, K A and J. Smith (eds.) *Toward a General Theory of Expertise: Prospects and Limits*, Cambridge University Press, Cambridge, UK.
- Krugar, C., Cross, N., (2006) Solution Driven Versus Problem Driven Design: Strategies and Outcomes. *Design Studies* 27(5): 527-548.
- Lawson, B. (1979) Cognitive Strategies in Architectural Design *Ergonomics* 22(1): 59-68.
- Liikkanen, L. & Perttula, M. (2008) Exploring Problem Decomposition in Conceptual Design Among Novice Designers. *Design Studies* 30(1): 38-59.
- Linsey, J.S., Wood, K.L., and Markman, A.B. (2008) Modality and representation in analogy. *Artificial Intelligence for Design, Analysis and Manufacturing* 22(02): 85-100.
- Lloyd, P., & Scott, P. (1994) Discovering the design problem, *Design Studies*, 15 (2): 125-140.
- Maher, M. L., Poon, J., & Boulanger, S. (1996) Formalising Design Exploration as Co-evolution: a Combined Gene Approach, in J S Gero and F Sudweeks (eds) *Advances in formal design methods for CAD*, Chapman and Hall, London, UK.
- Mak, T.W. & Shu, L.H. (2004). Abstraction of Biological Analogies for Design. *CIRP Annals – Manufacturing Technology* 53(1):117 – 120.
- Nagel, R., Midha, P, Tinsley, A, Stone, R., McAdams, D., Shu, L. (2008) Exploring the Use of Functional Models in Biomimetic Concept Design. *ASME Journal of Mechanical Design*, 130(2).
- Nagel, J., Stone, R., McAdams, D. (2010a) Exploring the Use of Category and Scale to Scope a Biological Functional Model. *Proc. ASME Int. Design Engineering Technical Conf.*, Montreal, Canada.

- Nagel, J., Stone, R., McAdams, D. (2010b) An Engineering-to-Biology Thesaurus for Engineering Design. *Proc. ASME Int. Design Engineering Technical Conf.*, Montreal, Canada.
- Nersessian, N. (2009) How do engineering scientists think? Model-based simulation in biomedical engineering laboratories, *Topics in Cognitive Science*, 1: 730-757.
- Newell, A. and Simon, H.A. (1972) *Human Problem Solving*, Prentice-Hall, Englewood Cliffs.
- Rittel, H., Webber, M. (1974) Dilemmas in a General Theory of Planning, *Policy Sciences*, 4: 155-169.
- Pahl, G., & Beitz, W. (1984) *Engineering design* (K. M. Wallace, Trans.). London: The Design Council.
- Reitman, W. R. (1964) Heuristic Decision Procedures, Open Constraints and the Structure of Ill-defined Problems. In M.W.Shelly & G.L. Bryan (Eds.) *Human Judgments and Optimality*, New York, Wiley.
- Restrepo, J and Christiaans, H (2004) Problem Structuring and Information Access. *Design Journal of Design Research* (4).
- Roozenburg, N.F.M. and Eekels, J. (1995) *Product Design: Fundamentals and Methods*. Wiley, Chichester.
- Rowe, P. G. (1987) *Design thinking*. Cambridge, MA: The MIT Press.
- Sarkar, P. & Chakrabarti, A. (2008) The Effect of Representation of Triggers on Design Outcomes. *Artificial Intelligence for Design, Analysis and Manufacturing*, 22(02): 101-116.
- Schön, D A (1983) *The Reflective Practitioner*, Temple-Smith, London, UK.
- Schön, D A (1988) Designing: Rules, Types and Worlds, *Design Studies*, 9: 181-190.
- Shu, L., Stone, R., McAdams, D, Greer, J. (2007) Integrating Function-Based and Biomimetic Design for Automatic Concept Generation. In *Proc. International Conference on Engineering Design (ICED'07)*, Paris, France, August 2007.
- Simon, H. A. (1973) The Structure of Ill-structured Problems. *Artificial Intelligence*, 4:181-201.
- Singh, L. H., Hansen, H. N., Gegeckaitė, A., Moon, J., and Chan, C. (2006) Case Study in Biomimetic Design: Handling and Assembly of Microparts, *Proceedings of the ASME IDETC/CIE*, Philadelphia, PA.
- Stone, R., Woods, K (2000) Development of a Functional Basis for Design. *ASME Journal of Mechanical Design*, 122(4).
- Vattam, S., Helms, M., and Goel, K. (2007) Biologically-Inspired Innovation in Engineering Design: A Cognitive Study. Technical Report, Graphics, Visualization and Usability Center, Georgia Institute of Technology GIT-GVU-07-07.
- Vattam, S., Helms, M., & Goel, A. (2009) Nature of Creative Analogies in Biologically Inspired Innovative Design. In *Proc. Seventh ACM Conference on Creativity & Cognition*, Berkeley, California
- Vattam, S., Wiltgen, B., Helms, M., Goel, A. & Yen, J. (2010) DANE: Fostering Creativity in and through Biologically Inspired Design. In *Proc. First International Conference on Design Creativity*, Kobe, Japan.
- Vattam, S., & Goel, A. (2011) Foraging for inspiration: Understanding and supporting the information seeking practices of biologically inspired designers To appear in *Proc. ASME IDETC Conference on Design Theory and Methods*, Washington DC.
- Vincent, J.F.V., Bogatyreva, O.A., Bogatyreva, N.R., Bowyer, A., and Pahl, A.-K. (2006) Biomimetics: Its Practice and Theory, *Journal of the Royal Society Interface*, 3:471-482.

- Wilson, J.O. (2008) A Systematic Approach to Bio-inspired Conceptual Design. PhD Dissertation. Georgia Institute of Technology, Atlanta, USA.
- Wilson, J.O., and Rosen, D. (2007) Systematic Reverse Engineering of Biological Systems. *Proc. 2007 ASME IDETC/CIE*, Las Vegas, Nevada.
- Wirth, N. (1971) Program Development by Stepwise Refinement. *Communications of the ACM*, 14(4): 221-227.
- Yaner, P. & Goel, A. (2006) Visual Analogy: Viewing Retrieval and Mapping as Constraint Satisfaction. *Journal of Applied Intelligence*, 25(1):91-105.
- Yen, J., & Weissburg, M. (2007) Perspectives on Biologically Inspired Design: Introduction to the Collected Contributions. *Journal of Bioinspiration and Biomimetics*, 2.
- Yen, J., Weissburg, M., Helms, M., Goel, A. (2011) Biologically Inspired Design: A Tool for Interdisciplinary Education. To appear in Yoseph Bar-Cohen (ed) *Biomimetics: Nature-Based Innovation*. CRC Press.

15. Terminology

Listed below are a number of similar but distinct terms and their definitions.

Design problem: The problem which the designer is given (or creates) which the designer must satisfy for the final design to be considered successful.

Design problem description: The design problem, as described by the designer, design team, and/or client. Problem descriptions in this work refer to text descriptions, static visual/graphical descriptions, and the accompanying annotations on such visual descriptions.

Design problem model: The underlying model of the design problem present in the mind of the designer. Such models are considered in this work to be sets of concepts and relationships among concepts.

Design problem schema (P-Schema): The abstract set of concepts and relationships from which a design problem model is composed.

Designer problem goal (DPG): A part of the design problem schema that represents a designer goal where the end state of the goal is to alter the current problem model.

Solution: Any design that meets one or more of the criteria established in the design problem.

New (or Proposed or Candidate) solution: A solution to a design problem that is currently in the process of being designed.

Existing solution: A solution that meets one or more criteria of the design problem and that exists prior to the start of the design process (e.g. an axe is an existing solution for cutting down trees).

Biological solution: An existing biological organism that could be construed to meet one or more criteria for solving a design problem (e.g. a beaver is a biological solution for cutting down trees).

Solution description: A solution as described by the designer or design team. Solution descriptions in this work refer to text descriptions, static visual/graphical descriptions, and the accompanying annotations on such visual descriptions.

Solution model: The underlying model of a solution that is implied to be present in the mind of the designer. Such models are considered in this work to be sets of concepts and relationships among concepts.

Solution schema (S-Schema): The abstract set of concepts and relationships from which a solution model is composed.

Solution-Problem [S, P] pair: The description of a combination that exists in memory of a problem model for some design problem and a solution model for a solution that meets one or more of the criteria for satisfying the problem. A solution model must be paired with one or more problems. A problem model may be paired with one or more solutions.

Problem concept: A single conceptual unit represented in the problem-schema.

Problem concept relationship: A relationship between two problem concepts.

Conceptual turnover: When comparing a problem model at time (T) to a model of the same problem at time (T+n), conceptual turnover refers to the number of concepts that exist in the model at time T but no longer exist in the model at time (T+n), as a ratio of the total number of concepts at time T.

Conceptual design: A design formulated usually at the level of functions, and the working principles to be used to realize those functions. Material and structural specifications are usually included only to a level of detail sufficient to show high-level system relationships, assess general feasibility, meet certain material criteria present in the design problem, and to aid in communicating the concept.

Appendix A: Problem-Driven Design Example

Sample problem-driven design project: i-Fabric

The i-Fabric project followed the problem-driven approach prescribed in class by the instructors. The design team arrived at the problem of conceptualizing “a thermally responsive and adaptive fabric that can be made into clothing in order to provide thermoregulation for the user in extreme temperature environments.” The problem was reframed, or “biologized,” as: “How are organisms in nature capable of maintaining consistent body temperatures using the least amount of energy?”

The designers found six different sources of biological inspiration, including arctic penguins, wood storks, arctic wolves, beehives, Kenyan chameleons, and humans. These choices illustrate the effectiveness of reframing the problem in biological terms. Not only do the biological entities come from different environments, they represent a wide range of behaviors at a number of different levels of biological organization, e.g. organism systems, to collections of organisms. Each source was evaluated, and an initial solution (Figure 2a) was selected based on the beehive, which uses the phase transition properties of a paraffin wax to store and release heat to moderate the temperature of the hive.

The designers initially limited themselves to making a composite material with wax. Because of the use of the same physical components from source to solution, rather than application of the principle, we classify this as a structurally focused project. As was common, the designers also encountered a new problem, in this case the problem of heat localization in the human body, and required a solution that could shift heat from central to peripheral body locations. New sub-problems, of which this is one example, occurred in all of the observed design problems. As solutions were tried, obstacles to their implementation (new requirements), deficiencies in the solution (partial-solutions, as in this case), or inspiration from the source analogue provided a deeper insight into the problem, and resulted in additional design iterations.

The second iteration of the design process yielded a current bypass system (Figure A-1) found in several other cases (wood stork and arctic wolf) that could redirect other parts of the body. Using this inspiration, the combined the phase transition material with heat conducting fibers that could channel heat from one of the body to another (Figure A-2). Note here the more away from the literal, structural translation of

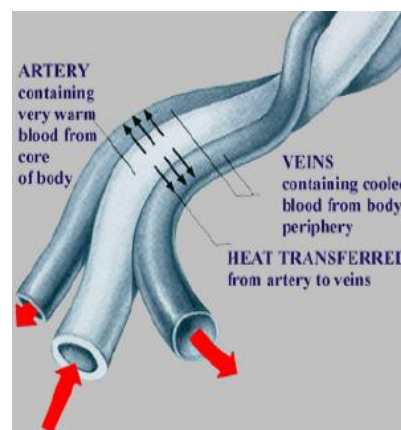
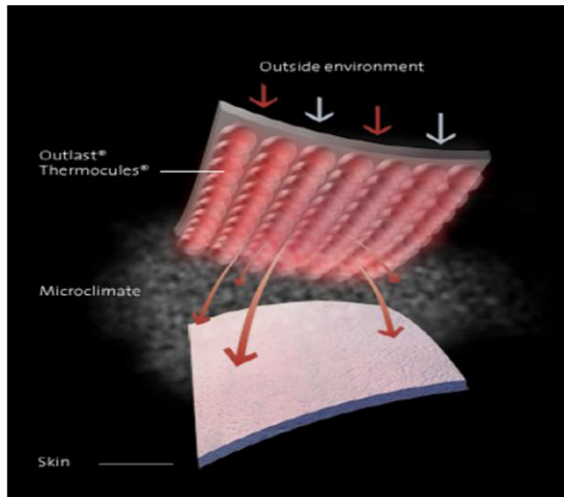


Figure A-1: Counter-current bypass system

counter-
biological
heat to
designers

location
designers
moving a

fluid, to moving only the essential substance, heat. Since the final design used two separate mechanisms, each to accomplish a separate function, we classify this as an example of *compound analogy*, but not as a multi-functional solution. Multi-functional solutions differ in that they use a single mechanism to accomplish two or more functions.



a) Initial Design Concept, from Beeswax



b) Final Design Concept, incorporating beeswax and counter-current systems.

Figure A-2: Problem-driven design trajectory for the i-Fabric project.

Appendix B: Solution-Driven Design Example

Sample solution-driven project: Abalone Armor

The abalone armor project provides an excellent example of the solution-driven approach. The team first determined that they wanted to use abalone shell (Figure 3), in particular, nacre, as their inspiration, and then formulated a problem that could be solved by the impact-resistant nacre. The designers had an initial understanding of the superficial characteristics of nacre, and quickly settled on the problem of conceptualizing a bullet-proof vest using the abalone nacre. Later, the team abstracted their problem specification to “using a material that combines the qualities of strength, toughness and self-healing”, and reframed their problem as the question: “What characteristics do organisms have that enable them to prevent, withstand and heal damage?”

Using these more abstract problem definitions, the students were instructed to investigate other sources. They looked at spider silk, lobster exoskeleton, sea star, rhino horn, and human bone. Each new alternative was dismissed after a short period of analysis, demonstrating a solution fixation that was common among all groups. That is, as soon as an initial biological source of inspiration was investigated, that source of inspiration tended to dominate all future solution development. Alternative sources of inspiration were dismissed as soon as a potential challenge was encountered, despite the fact that some of those same challenges also were true of the initial, fixated source of inspiration.

As the designers understood the behavior of the abalone shell better, an understanding developed that suggested to the designers that the way substances react to forces not only depended on the magnitude of the force, but also the duration. This created an elaboration of the problem to include resistance to both bullets and knife strikes, which apply different force magnitudes over different time frames. Because there were now multiple functions which the solution needed to address we classified this as a multi-functional problem.

The analysis of abalone nacre, including fracture mechanics of response to bullet impact based on criteria such as fracture stress, surface energy, strength intensity, and minimum initial crack size, showed that body armor made from abalone nacre would be one hundred times too weak to stop a bullet, and would weigh ten times more than conventional Kevlar body armor. The analysis in this project was mostly technical and quantitative, and did not result in a conceptual design.

All phases of design, initial through final, mimicked the components and materials of the abalone shell exactly, assuming the same composite of calcium carbonate and protein would be applied to the bullet-proof vest. Closely mimicking structure in this way was another common design practice highlighted by this case. Because

the final design attempted to use a single solution to meet the needs of the multi-functional problem, we classified this as a multi-functional solution.

Appendix C: Compound Analogy Design Example

Example of Compound Analogy: Eye in the Sea

The goal of this project was to design an underwater micro-bot with locomotion modality that would ensure stealth. The initial research for the underwater micro-bot focused on the copepod (a small crustacean, 1-2mm in length) as a source for understanding stealthy locomotion. In exploring this concept, designers became aware that the copepod used two distinct rhythms of appendage movement for achieving motion underwater. A slow and stealthy rhythm was used when foraging for food and a quick but non-stealthy rhythm was used when escaping from predators. This understanding led the designers to decompose their original problem into two separate functions, one for slow movement, and one for rapid movement, both of which required stealth. *This new problem decomposition was based solely on the understanding gleaned from the copepod analogy.* The knowledge of the slow, stealthy mechanism used by the copepod, known as a “metachronal beating pattern,” was transferred from the copepod source to create a partial solution to the problem.

Next, the designers had to address the second sub-function (fast, stealthy motion). They identified squid locomotion as an inspiration for achieving this function. The squid mechanism, jet propulsion, is both much *faster* and *stealthy* because its wake matches the external disturbances that naturally occur in the surrounding water. Notice the stealth achieved here (wake matching) is significantly different from the way stealth is achieved in copepod motion (wake minimizing).

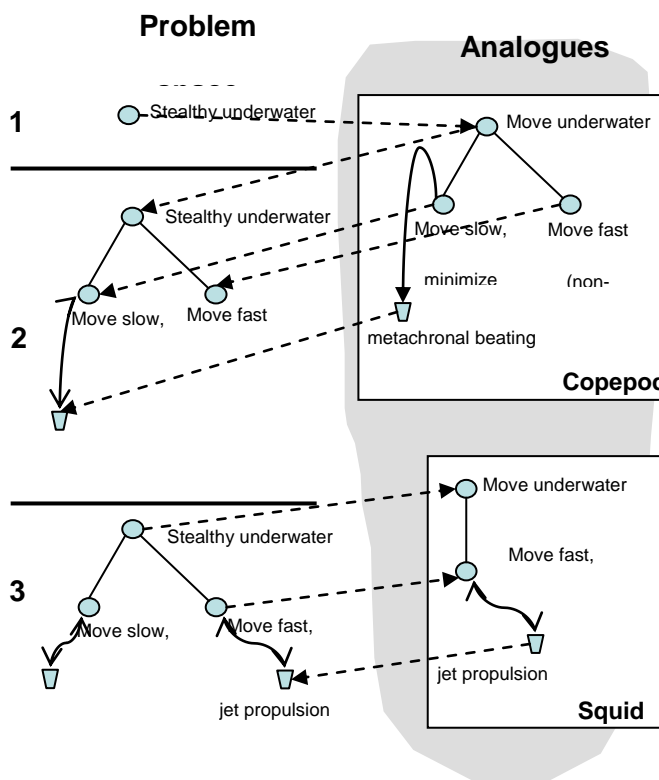


Figure C-1. Design Trajectory of the Eye in the Sea.

Fig. C-1 develops a model of the generation of this solution using the framework of the compound analogical design. Step 1 depicts the nature of the problem space early in the design. The main function is to move underwater stealthily, and the copepod is identified as a solution analogue. In Step 2, *based on knowledge from the copepod analogy*, the function of moving underwater is decomposed into sub-functions. The solution to the function of moving slowly by minimizing wake is adapted from the copepod to generate a partial solution. But the function of moving fast, yet stealthily remains unresolved in Step 2. In step 3, the analogue of squid is retrieved to address this function. Its solution of using jet propulsion for movement is transferred to the current problem to generate the remaining solution. These two partial solutions are aggregated to achieve the trial design.

Appendix D: Instructor Requirements List

Requirement	Category
Provide a structure for evaluating the fitness of analogies	analogy
Provide a means for understanding good and bad levels of analogy	analogy
Allow for differences in analogical context to be made specific	analogy
Motivate all students to contribute equally	comm
Record individual student inputs.	comm
Force students to record their interpretation of team meetings.	comm
Enable analogies to be shared among team members	comm
Enable graphic representations to be modified by more than one person	comm
Force the use of graphic representations	comm
Supports the BID process directly	ds
Track changes to solutions over time.	ds
Force students to consider current solutions.	ds
Force students to consider how their solution will be superior.	ds
Show the hierarchies of problems and solutions	ds
Enable systems to be represented graphically (boxes and arrows)	ds
Allow for a “dialogue” between problems and solutions	ds, prob
Help students better define problem	prob
Ask students questions about their problem.	prob
Force students to define problem in terms of final results	prob
Define design sub-problems as they occur	prob
Create traction for vaguely defined, intractable problems.	prob
Assist students in problem partitioning	prob
Assist students in evaluating problem partitions	prob
Define measures of importance for problem partitions	prob
Define problem (not solution) inputs and outputs.	prob
Provide a means for understanding problems at different levels of abstraction	prob
Provide a means to describe for natural systems the problems in nature they are solving	prob
Allow for problems to be changed to better match solutions	prob
Catalogue problems to solve	prob
Identify/describe “champion adapter” solutions	search
Do not undermine student’s ability to learn search skills.	search
Applicable to undergrad design students	tech
Deployed within a classroom context	tech

Appendix E: Problem Schema Specification

Artifact Specifications

The term “artifact specifications” defines properties and values, quantitative and qualitative, related directly to the artifact being designed. Such specifications may be “limitations”, designating properties and values that include absolute terms of inclusion or exclusion such as “must”, “cannot”, or “should”. Other specifications denote “options”, which make explicit possibilities for properties and values that may associated with the design artifact, expressed in terms like “could”, “might”, or “possibly” to name a few. Whereas a limitation is a statement such as “the design must use lightweight materials”, an option may be “the design could use lightweight metal foam”; both talk about a property material from which the artifact will be manufactured, however one express a condition of inclusion, while the other provides an alternative.

Artifact specifications can be categorized as applying to either manufacturing or performance aspects of the problem. Additional sub-types include, but are not limited to: temporal, energy, informational, material, cost, user, and structural.

Artifact specifications assume one or more aspects of the design solution. Often such assumptions are very high level e.g. that some material in the solution will interact with the operational environment. Such high level solution assumptions are rarely made explicit, and typically are not associated with a specific solution per se. For instance, solution assumptions may be applied to a class of proposed solutions (or sub-solutions), e.g. for the class of potential solutions that use combustion engines a design limitation may be “must use renewable biofuels.” This limitation would not apply to other classes of solutions that do not use combustion engines.

Operational Environment

Operational environment defines aspects of where, when, with whom and under what conditions the artifact will operate, including: location, condition, reactive entities, people/users, and time. We find in our context, this excludes aspects of the manufacturing environment per se, but it need not. In such a case, additional sub-types, such as manufacturing process, may emerge.

Unlike artifact specifications which define properties and values, operational environment concepts are often conceptual indexes to what appear to be rich mental models of environments. Often there references highlight salient aspects of the environmental model in question. An example of an operational environment might be “under water,” specifying that that the design solution must perform

some or all functions while immersed in water. The designer may then draw attention to specific, salient aspects of the environment model, for example including range values, such as “under fresh water and salt water between 10’ and 1000’ of ocean depth”.

Cognitively, operating environments may or may not be tied to more complex models or experiences of the underlying operational environment entity. In the above example of “under water”, models of water, liquids, pressure, salinity, dissolved gasses, temperature gradients, and other complex aspects of the environment may or may not be available to the designer, depending on the level of expertise and complexity of the underlying mental model. Furthermore, the designer will have varying degrees of personal experience with the environment (episodic memories), as well as varying degrees of knowledge about other entities (solutions) that have operated in that or similar environments. Thus operational environments and environmental features that are rendered salient may also serve as indexes into additional environment models, designer experiences, and existing solutions. Models and schema for operational environments, as well as the process by which such model pointers may be used during design is beyond the scope of this proposal.

Function

Most engineers and designers are already familiar with the concept of function. Simon frames function as the interface between the internal and external environments, function basis models frame function in terms of changes to (inputs and outputs) of substances flowing through a system, while still others view function as the intended result of the system. In the context of a design episode, we consider functions as those actions which the intended device must perform. This takes the form of some (subject, verb, noun) tuple, where the subject is the to-be-designed artifact, the verb reflects the action in question, and the noun reflects the object on which the subject is acting, which we shall call function-object. In many cases, the noun is the same as the subject e.g. to move self, to clean self, etc. In other cases multiple function-objects are required; especially for cases of change e.g. device changes heat into electricity.

Functional hierarchy is prevalent in design theory and prescriptive design techniques. The most commonly discussed hierarchy is the system/sub-system hierarchy, in which the function of the sub-system contributes to the function of the larger system. Thus system S_1 performing function F_1 is comprised of sub-systems $S_{1-1}, S_{1-2}, S_{1-n}$, which perform sub-functions $F_{1-1}, F_{1-2}, \dots, F_{1-n}$. Each sub-system can then recursively be defined by additional sub-systems and sub-functions. In this sense the functions can be seen as additive, or AND-type conjunctions. In order to accomplish F_1 , the system must perform all

sub-functions F_{1-1} AND F_{1-2} AND...AND F_{1-n} . In design thinking, designers also consider multiple alternative functions, which we consider OR-type conjunctions. Chandrasekaran provides an excellent description of such mixed AND-type/OR-type function hierarchies, including the implications of such mixed hierarchies on computational search.

Function relationships may be hierarchical, but not always decompositional. The number of child-nodes for a parent need not be (and often are not) greater than one. In practice, we see such functional abstractions expressed in terms of a function (F_{1-1}) in the service of another higher-level function (F_1). In the example, “we can inhibit activity in order to self-regulate, which helps us save resources” we see a three tier relationship where F_{1-1-1} (inhibit activity) is in the service of F_{1-1} (regulate self) which is the service of F_1 (saving resources).

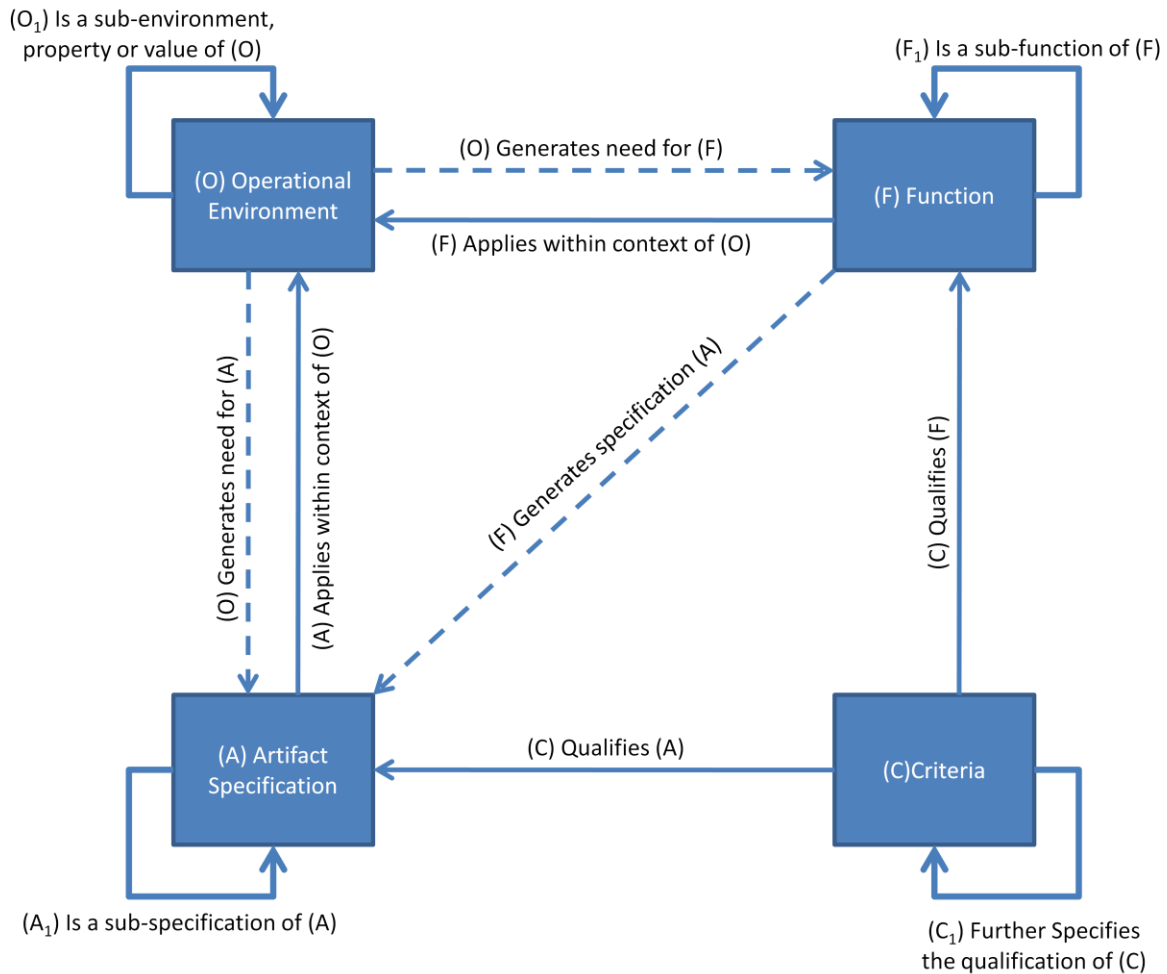
While we do observe instances of system/sub-system functional hierarchies, often imposed by prescriptive methods of design, in unstructured innovative design practice we observe much less rigid organizational structures. In general, the verb and function-object aspects of the concept seem to be only loosely coupled, such that verbs and function-objects can be combined and recombined throughout extended design episodes e.g. (store heat) and (protect from oxidation) can later become (protect from heat).

Criteria

This fourth category, which we call criteria, provides additional information against which to measure success for an already (or to-be) defined aspect of the problem, usually functional but occasionally related to a design specification. Such criteria are established in terms of either an *absolute* criteria e.g. the system must convert light to electricity with 32% efficiency, or *relative* to some existing solution e.g. the system must convert light to electricity more efficiently than existing photovoltaic cells. In problem models early in the design process we find that such criteria are often very vague, such as “the new design must be more efficient than existing designs.” In this case the criteria “more efficient” might apply to any aspect of the new design. More efficient use of power, movement, manufacturing processes, etc., may be included in such a generalized criteria. Despite the seemingly vague character of some of these criteria, we find them used frequently.

Problem Schema Relationships

Individual problem elements often exist in relation to other conceptual elements. Function/sub-function relationships are one example, as are criteria, which almost always further describe some other existing descriptive element (often times implicitly). Figure E-1 provides a summary of relationships



among design description elements, excluding problem-solution relationships. Solid lines represent explicit relationship types, often explicitly expressed in problem descriptions. Dotted lines represent some process relationship, expressing that the concepts are related through some cognitive process in which the generation of one is linked to the other. Other than to note this relationship exists, we offer no additional elaboration in this paper on these processes.

Appendix F: Designer Problem Goals

A designer problem goal is an designer objective that, for some given problem state at time t , $P(t)$, transforms the problem state to a new state $P(t+1)$. The type of transformation goal is tied to the type of challenge the designer is current facing (e.g. dynamics, complexity, lack of knowledge, etc). The most straightforward kind of transformation is the concept addition transformation. This involves adding a concept and zero or more concept relationships to the problem schema. On the other hand, two ways of dealing with complexity, for example, are to (1) decompose the problem, or (2) to partition the problem. Decomposing involves adding new sub-concepts to existing concepts; whereas partitioning involves the separation of existing sets of concepts into sub-sets that can be considered independently.

The specification for a specific designer problem goal (DPG) describes the objective of the goal (in plain English) and provides a definition of pre- and post- problem states in abstract terms. For instance “and” decomposition, shown in Figure F-1, involves the addition of one or more sub-concepts to an existing concept, grouped by an and conjunction, the implication of which is that in order to fulfill the condition of the first concept, the conditions of all sub-concepts must be fulfilled.

“And”-type decomposition

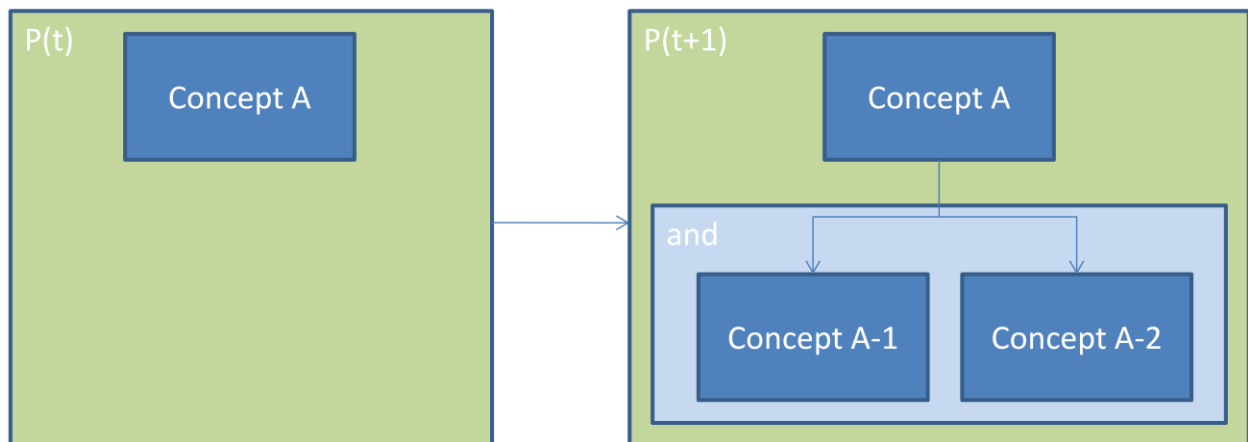


FIGURE F-1. PROBLEM SCHEMA CONCEPTS AND RELATIONSHIPS